



МІНІСТЕРСТВО  
ОСВІТИ І НАУКИ  
УКРАЇНИ РЕКОМЕНДУЄ



#### 4 мови програмування



Управління  
звуком, світлом,  
електродвигуном



Використовуємо  
віртуальний  
осцилограф



Живлення від  
батарейок або  
від мережі

Та інші цікаві проекти

ВІД 8 ДО 98 РОКІВ

граємо та вчимося

# ЕЛЕКТРОННИЙ КОНСТРУКТОР **ZNATOK™** для Arduino•START

python™  
GRAPH  
SCRATCH C

Управління звуком, світлом, електродвигуном

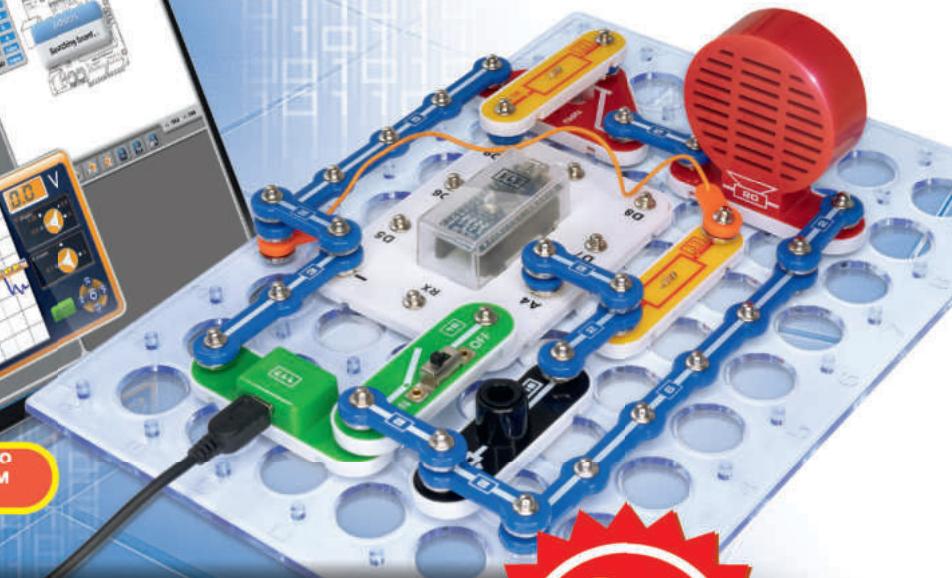
Використовуємо віртуальний осцилограф

Живлення від батарейок або від мережі

Рекомендовано для дітей віком від 8 років

9 ПРОЕКТІВ!

• ОРИГІНАЛЬНИЙ СПОСІБ З'ЄДНАННЯ ДЕТАЛЕЙ!  
• ЛЕГКЕ ЗБИРАННЯ  
• НЕ ТРЕБА ПАЯТИ!  
• ШВІДКИЙ РЕЗУЛЬТАТ  
• ДЕТАЛЬНИЙ ОПИС



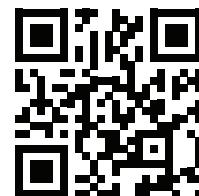
Конструктор можна вважати початком вивчення та пізнавання принципів роботи мікроконтролерів на прикладі Arduino Nano та першим кроком до робототехніки. У наборі реалізована можливість як самостійного вивчення, так і сумісної роботи з педагогом. Під час роботи з цим набором можна використовувати 4 мови програмування – найпростішої Graph Z до мови C++, якою користуються професіонали.



Це справжня допомога в освіті.

Будь ласка, перш ніж розпочати виконання проектів, уважно прочитайте інформацію в розділах «ВСТУП», «ОПИС ДЕТАЛЕЙ» та «ВСТАНОВЛЕННЯ ПРОГРАМ».

QR-коди, що присутні в керівництві, дозволять побачити роботу схеми або краще зрозуміти принципи роботи.



Щоб побачити всю лінійку конструкторів ZNATOK, відскануйте QR-код:



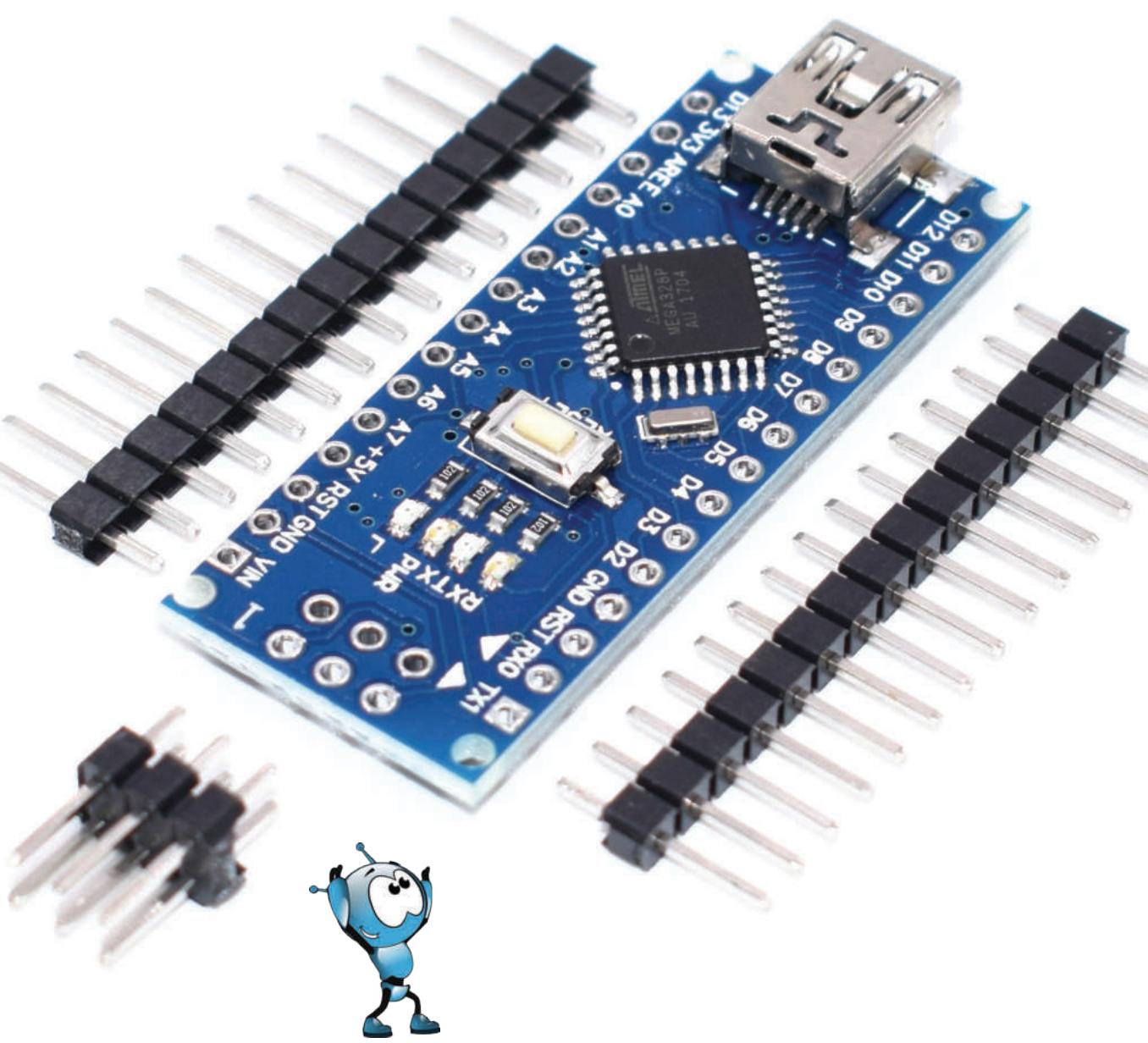
Перед початком гри рекомендуємо уважно ознайомитись з інструкцією\* та деталями конструктора.

\* Адаптація інструкцій ZNATOK узгоджена з викладачами фізики, аби уся термінологія українською була подана вірно.

\*\* Конструктори ZNATOK схвалено Міністерством освіти України для практичних занять з фізики для учнів 8, 9, 10, 11 класів та рекомендовано для використання в ігрових цілях для дітей віком від 5 років.

<b>ВСТУП</b>	4
<b>ПЕРЕЛІК ДЕТАЛЕЙ</b>	6
<b>ОПИС ДЕТАЛЕЙ</b>	
• Дроти	8
• RGB-світлодіод	8
• Вимикач	9
• Фоторезистор	9
• Світлодіоди	10
• Динамік	11
• Кнопки	11
• Резистори	12
• Лампа розжарювання	12
• Електродвигун	13
• Модуль Ардуїно	14
• Змінний резистор	16
• Джерела живлення	18
• Транзистори	19
• Монтажна плата	19

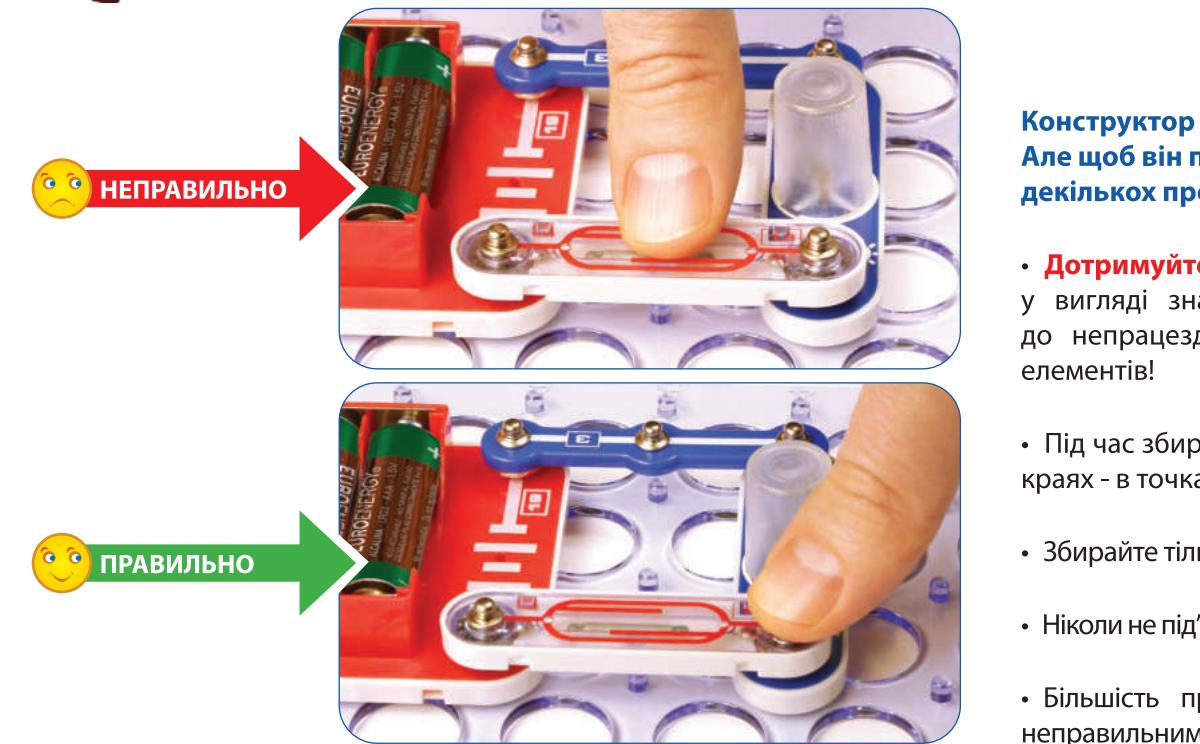
<b>ОПИС МОВ ПРОГРАМУВАННЯ</b>	20
<b>ВСТАНОВЛЕННЯ ПРОГРАМ</b>	22
<b>ПРОЄКТИ</b>	
• Керування світлодіодами	28
• Змішуємо кольори	46
• Світло та музика	52
• Керування електродвигуном	54
• Керування лампою розжарювання	58
• Синтезатори звуків	62
• Сигналізації	67
• Осцилограф	68
• Розумне світло	72
<b>ЦІКАВОСТІ</b>	
• Точність – чемність королів	74
• Ноти – це частоти	75
• Що таке RGB	76



## ВСТУП



Практично будь-який сучасний пристрій має мікроконтролери, які керують цим пристроєм та здійснюють зв'язок або з людиною, або з іншою технікою. Не кажучи вже про робототехніку, яка неможлива без мікроконтролерів. У даному наборі Ви навчитеся працювати з Arduino – модулем з популярного сімейства мікроконтролерів AVR. Отримані знання допоможуть Вам у майбутньому створювати нові моделі роботів, побутової техніки, комп’ютерів, смартфонів.



**Конструктор повністю безпечний та простий у використанні. Але щоб він працював якомога довше, треба дотримуватись декількох простих правил:**

- Дотримуйтесь полярності!** Деякі елементи мають маркування у вигляді знаку «+». Недотримання полярності призводить до непрацездатності схеми або пошкодженню електронних елементів!

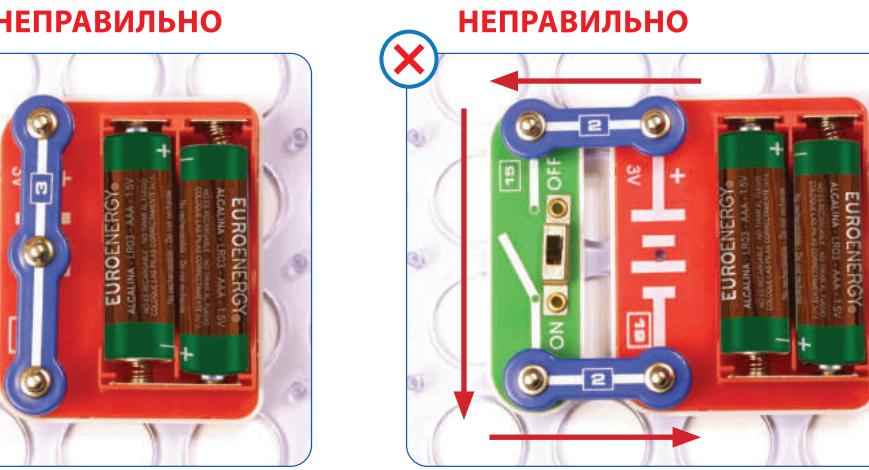
- Під час збирання схеми натискайте не на середину деталі, а по краях - в точках кріплення.
- Збирайте тільки ті схеми, які описані в даному керівництві.
- Ніколи не під’єднуйте схеми до електричних мереж Вашого будинку.
- Більшість проблем в електрических ланцюгах пов’язані з їх неправильним збиранням. Завжди уважно перевіряйте, що зібране

## ВСТУП

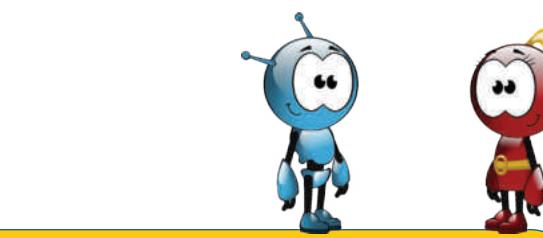
Вами коло дійсно відповідає тому, що зображене на рисунку. Переконайтесь, що всі з’єднання надійно закриті.

- Завжди вимикайте батареї, якщо якийсь елемент схеми почав сильно нагріватися.
- НЕ ДОПУСКАЙТЕ КОРОТКОГО ЗАМИКАННЯ БАТАРЕЙ!**

Ніколи не робіть так, як зображене на прикладах нижче та інших схожих випадках – коли выводи «+» і «-» з’єднуються напряму за допомогою дротів або вимикача. Завжди повинне бути якесь навантаження - світлодіод, резистор, електродвигун.



**Виробник не несе відповідальності за пошкодження деталей, які викликані неправильним з’єднанням деталей та недотриманням техніки безпеки.**



**СПОДІВАЄМОСЯ, що конструктор принесе Вам радість та допоможе краще пізнати приголомшлий світ електроніки!**

**ПЕРЕЛІК ДЕТАЛЕЙ**

Код	Найменування елемента	Кількість	Зображення
1	Дріт з 1 контактом	4	
2	Дріт з 2 контактами	9	
3	Дріт з 3 контактами	5	
4	Дріт з 4 контактами	2	
5	Дріт з 5 контактами	2	
6	Дріт з 6 контактами	1	
7	Дріт з 7 контактами	1	
15	Вимикач	1	
16	Фоторезистор	1	
17	Червоний світлодіод	1	

Дизайн деяких деталей з набору може відрізнятися від зображених у таблиці, але це не впливає на працездатність схем.

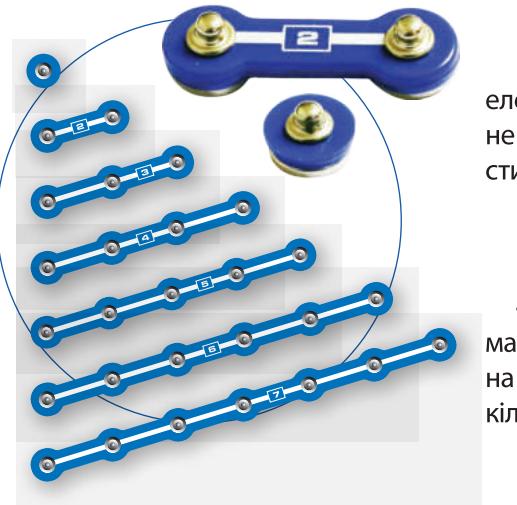
Код	Найменування елемента	Кількість	Зображення
20	Динамік	1	
26	Зелений світлодіод	1	
30	Резистор 100 Ом	1	
32	Резистор 5,1 кОм	1	
33	Резистор 10 кОм	1	
34	Резистор 100 кОм	1	
37	Лампа розжарювання	1	
38	Електродвигун	1	
52	Транзистор NPN	1	

**ПЕРЕЛІК ДЕТАЛЕЙ**

Код	Найменування елемента	Кількість	Зображення
111	Модуль Arduino Nano	1	
112	Синій світлодіод	1	
113	Жовтий світлодіод	1	
115	RGB-світлодіод	1	
116	Відсік для елементів живлення 4x1.5В (батареї не входять до комплекту)	1	
117	Модуль мережевого USB-адаптера	1	
118	Змінний резистор 5 кОм	1	

Найменування елемента без коду	Кількість	Зображення
Монтажна плата	1	
Кнопка маленька	2	
Дріт гнучкий з двома клемами	1	
Пропелер	1	
Прозорий циліндр	1	
Тримач	1	
Кабель USB-miniUSB	2	
USB-флеш-накопичувач (флешка)	1	

## ДРОТИ



**Електричні дроти** призначені для з'єднання елементів схеми між собою і не впливають на характеристики кола.

- **ЖОРСТКІ** сині дроти мають позначення, яке вказує на довжину цього дроту та кількість контактів на ньому.

Одиночні дроти-шайби призначені для електричного з'єднання між собою шарів схеми. У рідких випадках – для надання конструкції стійкості як «ніжки».

На схемах такі дроти-шайби непомітні, і тому місця, де вони повинні бути розміщені, позначаються ось таким чином:

- **ГНУЧКІ** дроти, які з обох боків мають контакт-кнопку для під'єднання до деталей схеми та жорстких дротів.



## RGB-СВІТЛОДІОД

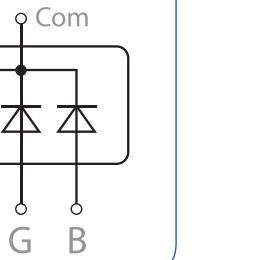


**RGB** — це перші літери англійських слів **Red-Green-Blue** (**Червоний-Зелений-Синій**). На відміну від звичайного світлодіода, тут всередині пластикової лінзи розміщені одразу три світлодіоди — **червоний**, **зелений** та **синій**.

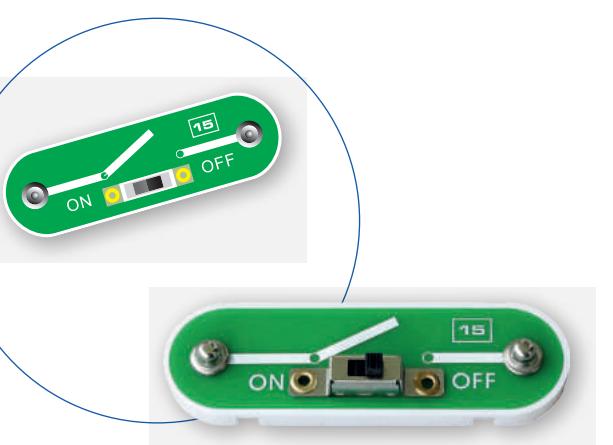
До модуля **115** дається прозорий циліндр, який вставляється у тримач.



**RGB-світлодіод** з цього набору має схему зі спільним катодом (захисні резистори не зображені).



## ВИМИКАЧ

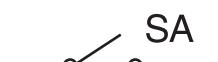


Вимикач має два положення – замкнено (**ON**), коли струм протікає через вимикач, та розімкнено (**OFF**), коли вимикач розриває ланцюг та струм не протікає через нього. Вимикач не впливає на характеристики ланцюга, але може змінити його функції, відключаючи або підключаючи до схеми елементи.

На електричних схемах вимикач позначається ось таким чином:



або



## ФОТОРЕЗИСТОР



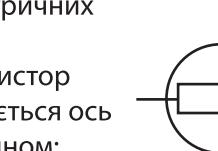
У фоторезистора з даного набору опір в темряві становить більше 20 МОм, а при дуже яскравому освітленні – 150 Ом.

Його ще називають світлоочутливим резистором – напівпровідниковий елемент, опір якого залежить від освітлення. Під дією світла опір фоторезистора зменшується. При цьому сила струму в колі збільшується та досягає значення, достатнього для увімкнення або роботи будь-якого пристрою.

## Галузь застосування

Датчики руху людини, протипожежні системи, вимірювачі температури (за яскравістю накалювання), автоматичні ліхтарі, охоронні сигналізації, вимірювачі освітленості та ін.

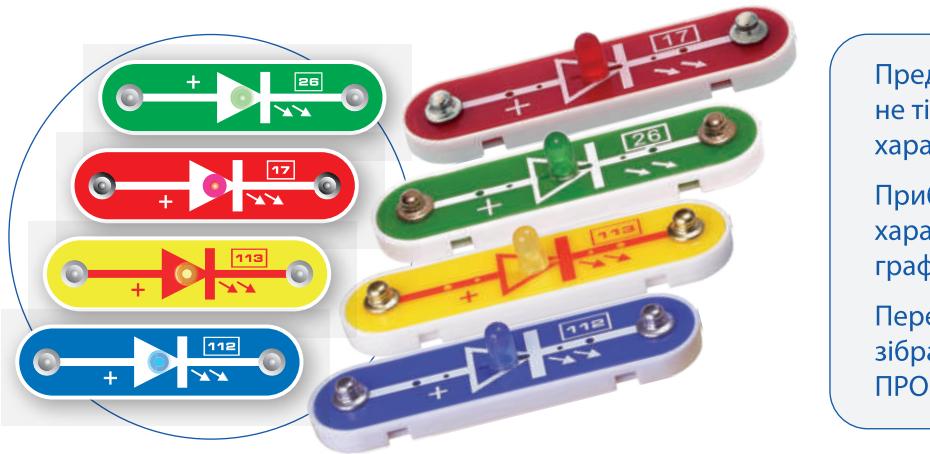
На електричних схемах фоторезистор позначається ось таким чином:



Якщо Ви бажаєте більше дізнатися про фоторезистори, це можна зробити, придбавши **Конструктор - ЗНАТОК - Школа (999+ схем)** -

**Практичне Заняття №17 «Фоторезистор»**  
подробиці на сайті <https://kiddisvit.ua/ua/brands/znatok>

## СВІТЛОДІОДИ



Представлені світлодіоди мають не тільки різний колір, але й різні характеристики.

Приблизні вольт-амперні характеристики (ВАХ) наведені на графіку.

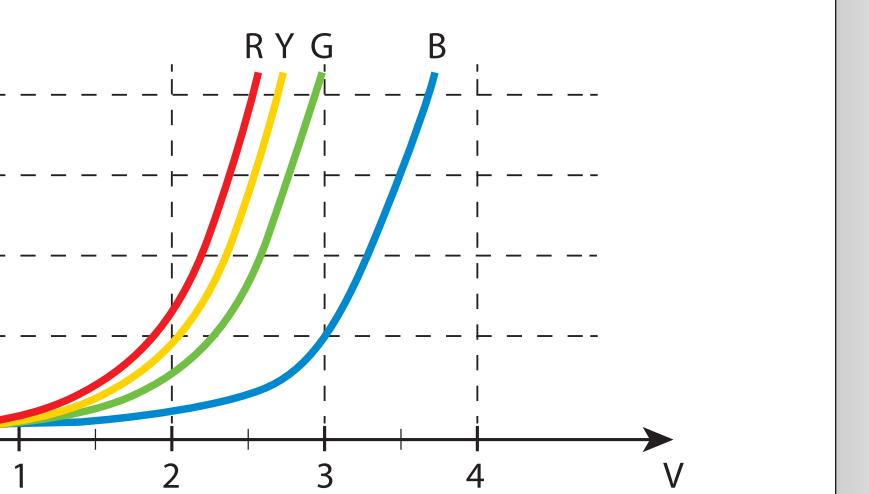
Переконатися в цьому можна, якщо зібрати та дослідити схеми в розділі ПРОЄКТИ.

Світлодіод (світловипромінюючий діод або LED - Light Emitting Diode) – це не лампочка, це електронний напівпровідниковий пристрій з р-п переходом, який починає світитися під час проходження через нього електричного струму.

**Під час підключення необхідно дотримувати полярності!**

Під час включення світлодіода потрібно використовувати струмообмежувальний резистор. У світлодіоді з цього набору такий резистор вже вбудований у корпус, тому в деяких схемах можливе підключення безпосередньо до джерела живлення.

На електричних схемах світлодіод позначається ось таким чином:

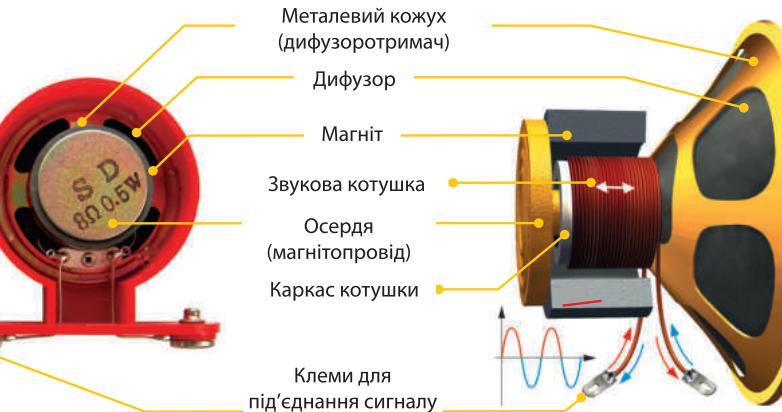
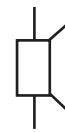


## ДИНАМІК



Динамік перетворює електричні коливання на звукові коливання. Динаміки можна знайти в навушниках, телевізорах, комп’ютерах, музичних колонках, телефонах тощо. Динамік ще називають гучномовцем.

На електрических схемах динамік позначається ось таким чином:



Якщо Ви бажаєте більше дізнатися про динамік, це можна зробити, придбавши Конструктор - ЗНАТОК - Школа (999+ схем) (подробиці на сайті <https://kiddisvit.ua/ua/brands/znatok>)

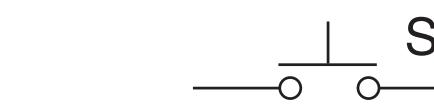
## КНОПКИ



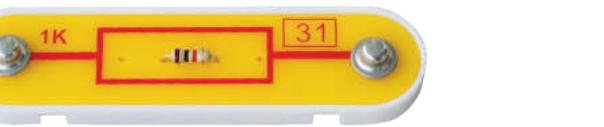
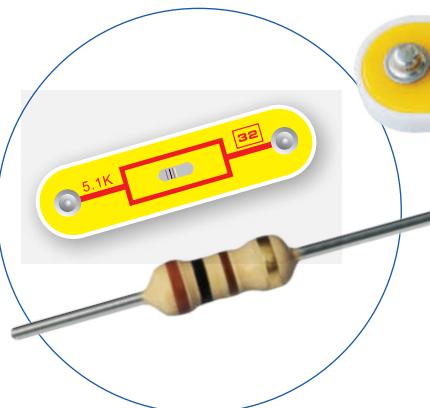
Кнопки бувають різних форм і розмірів, але всі вони пропускають струм, тільки коли на них натиснути – як на дверний дзвінок або кнопку в ліфті.

Кнопки не впливають на характеристики електричного ланцюга, але можуть змінити його функції, відключаючи або підключаючи до кола елементи.

На електрических схемах кнопка позначається ось таким чином:



## РЕЗИСТОРИ



**Резистор** (від латинської *resisto* – чиню опір) – елемент, який чинить опір електричному струму, встановлюється з метою регулювання струму та напруги. Чим вищий опір резистора, тим менший струм по ньому протікає.

Змініть резистори в схемі, і Ви побачите, що чим менший опір резистора, тим яскравіше світиться світлодіод.

При підключені резистора 100к світлодіод буде ледве «жевріти».

Опір вимірюється в омах (Ом або Ohm) з префіксом-множником «кіло», «мега», «гіга» і т.д. (Ом, кОм, МОм, ГОм... =  $\Omega$ , k, M, G,...).

$1000 \text{ Ом} = 10^3 \text{ Ом} = 1 \text{ кОм}$ ,

$1000 \text{ кОм} = 10^6 \text{ Ом} = 10^3 \text{ кОм} = 1 \text{ МОм}$

**Основні характеристики:** номінальний опір, точність, потужність розсіювання, температурний коефіцієнт, спосіб встановлення на платі.

**Галузь використання:** практично в усіх електричних ланцюгах. Якщо не у вигляді елемента, то всередині інтегральної схеми.

На електрических схемах резистори можуть позначатися двома різними способами:



Якщо Ви бажаєте більше дізнатися про резистори, це можна зробити, придбавши **Конструктор - ЗНАТОК - Школа (999+ схем)** (подробиці на сайті <https://kiddisvit.ua/ua/brands/znatok>)

Не можна довго дивитися на лампу, що світиться!

## ЛАМПА РОЖАРЮВАННЯ



Причини праці лампи розжарювання такі: під час проходження електричного струму через вольфрамову нитку вона нагрівається та починає світитися. Нитка знаходитьться всередині герметичної скляної колби.

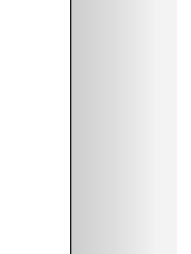
Лампа з цього набору розрахована на роботу з напругою не вище 6 вольт та максимальним струмом 0,3 ампера.

Незважаючи на активне впровадження світлодіодних ламп, лампи розжарювання продовжують випускати тому, що в них є ряд переваг, яких інші лампи не мають.

На електрических схемах лампа розжарювання позначається ось таким чином:



У цьому наборі лампа закрита напівпрозорим ковпачком. Без ковпачка вона виглядає ось так:



## ЕЛЕКТРОДВИГУН



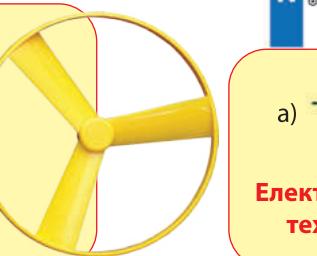
**Електродвигун** перетворює електричну енергію на механічну. В нашому випадку – електричну енергію на обертання валу. Чим вища напруга, що прикладається до електродвигуна, тим вища швидкість його обертання. Але занадто велика напруга може привести до пошкодження електродвигуна.

Існують електродвигуни постійного та змінного струму. Електродвигун **38** є двигуном постійного струму. Його ще називають електромотором або просто моторчиком.

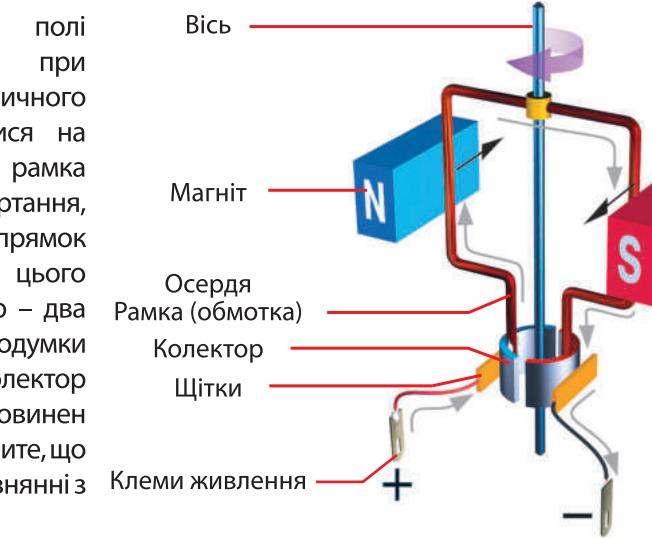
На електрических схемах він позначається ось таким чином:



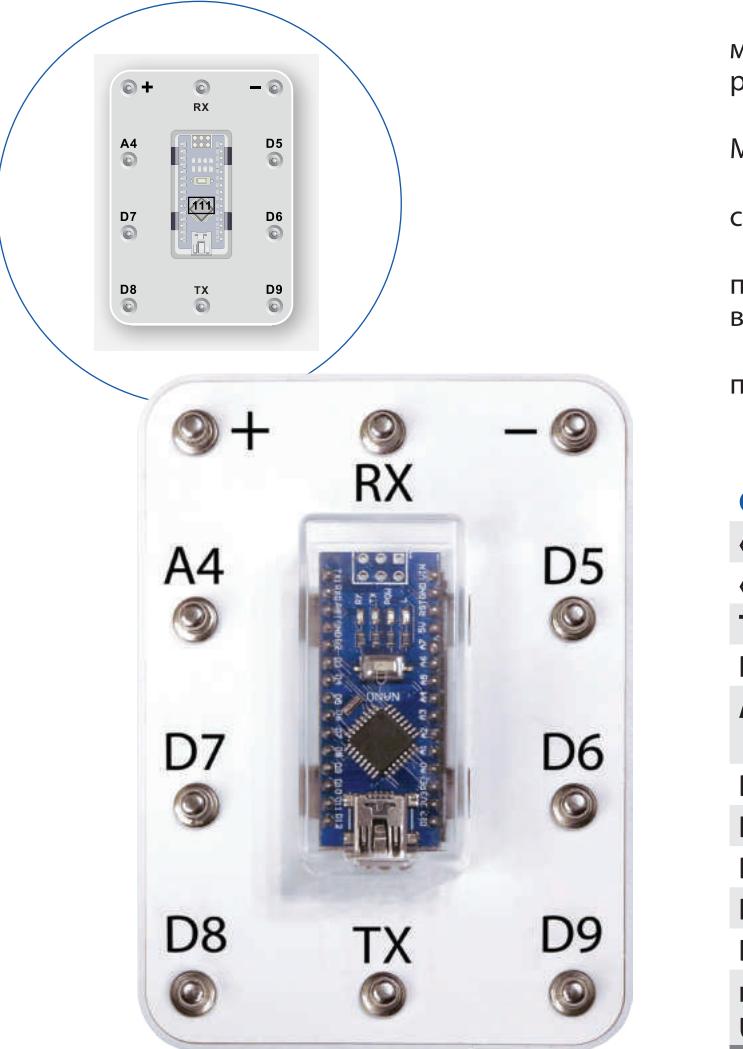
На валу двигуна з цього набору є насадка для встановлення пропелера.  
**!!! Під час встановлення пропелера на двигун НЕ ТИСНІТЬ на нього, а обережно покладіть пропелер на насадку.**



Електродвигуни є майже в кожній техніці: у роботах, автомобілях, літаках, побутовій техніці, комп’ютерах і навіть смартфонах, де вони беруть участь у вібродзвінку.



## МОДУЛЬ ARDUINO



Модуль 111 створений на базі **Arduino Nano**, який, в свою чергу, складається з мікроконтролера ATMEGA 328P сімейства AVR та додаткових елементів, необхідних для роботи.

Мікроконтролери бувають різні та можуть виконувати задачі як маленький комп’ютер. Мікроконтролер з цього набору може виконувати такі функції:

Видавати інформацію. За заданою програмою керувати зовнішніми пристроями – світлодіодами, лампами, електродвигунами. Відтворювати звуки, показувати час.

Приймати інформацію. За заданою програмою оброблювати інформацію від зовнішніх пристройів та відтворювати її на індикаторах. У цьому режимі можливе створення різних вимірювальних приладів.

Приймати інформацію, аналізувати її та на основі аналізу видавати інформацію. На цьому принципі створено чимало побутових приладів та роботів.

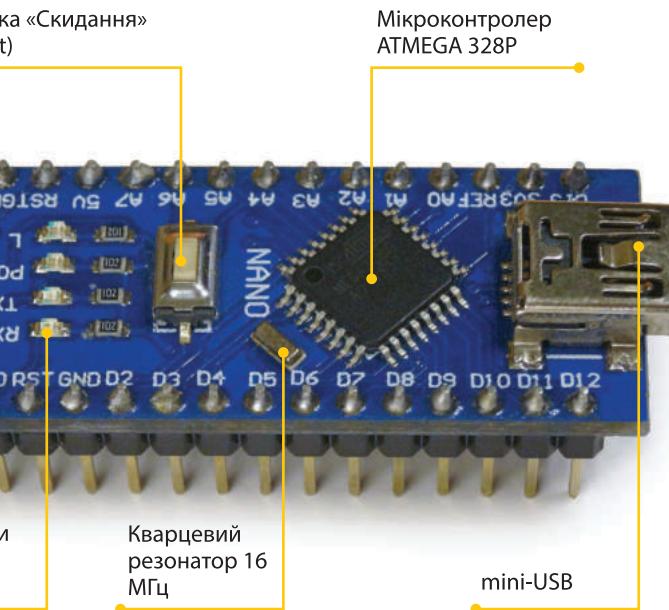
## Опис виводів модуля 111:

«+»	Плюс живлення
«-»	Мінус живлення
TX	Передача асинхронних вихідних даних
RX	Приймання асинхронних вихідних даних
A4	Аналоговий вхід. По суті, це вхід аналого-цифрового перетворювача (АЦП), який перетворює вхідну напругу на коди.
D5	Цифровий вхід/вихід з функцією ШІМ
D6	Цифровий вхід/вихід з функцією ШІМ
D7	Цифровий вхід/вихід
D8	Цифровий вхід/вихід
mini-USB	Роз’єм для з’єднання з комп’ютером

## ЯК ЦЕ ПРАЦЮЄ

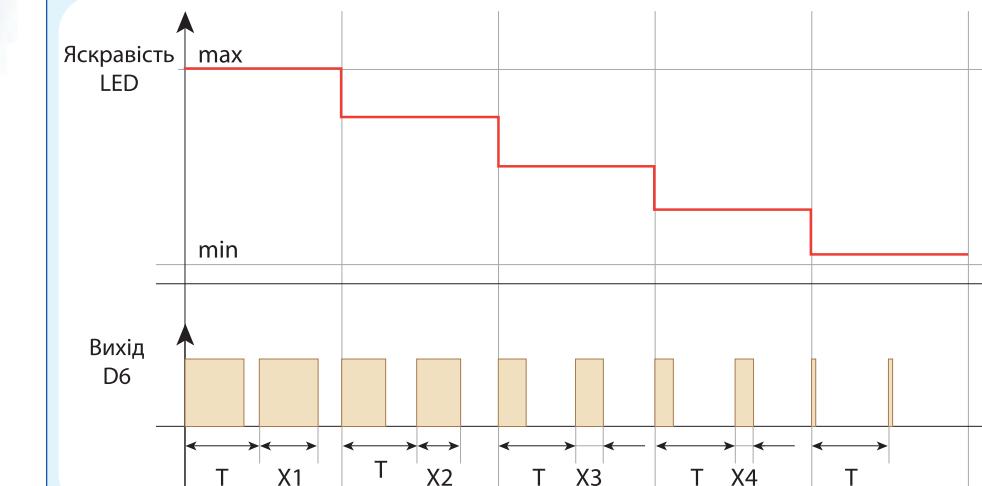


Широтно-Імпульсна Модуляція (ШІМ) англійською Pulse-Width modulation (PWM) — процес керування потужністю, яка підводиться до навантаження, шляхом прогальності імпульсів при постійній частоті.

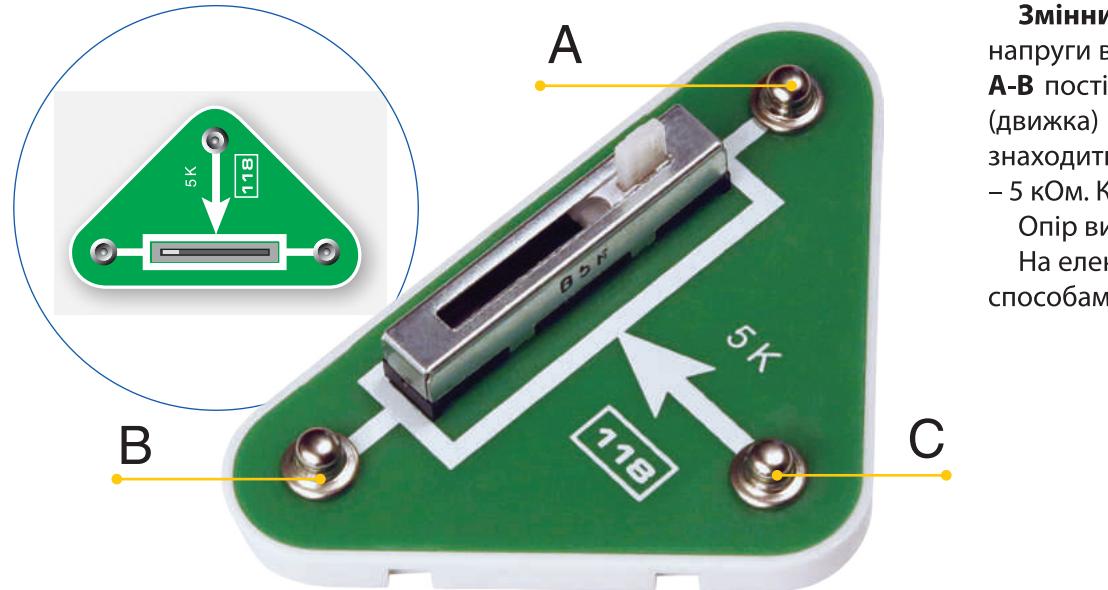


Як видно на фотографії, у модулі 111 використовуються далеко не всі виводи **Arduino Nano**. Але якщо Ви опануєте та зрозумієте принцип роботи тих виводів, що наявні на модулі 111, Ви легко зможете використовувати всі інші виводи у Ваших проектах.

**Прогальність (S)** – відношення періоду імпульсів T до тривалості імпульсу X, тобто  $S=T/X$ . У сигнала типу «меандр» прогальність дорівнює 2, тому що період у два рази більший, ніж тривалість імпульсу.



## ЗМІННИЙ РЕЗИСТОР



## Основні характеристики

Номінал опору, роздільна здатність, потужність розсіювання, температурний коефіцієнт, спосіб зміни опору.

## Галузь застосування

Змінні резистори застосовуються для регулювання яскравості світла, гучності звуку, швидкості обертання тощо.

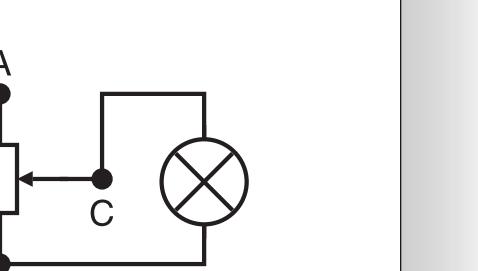
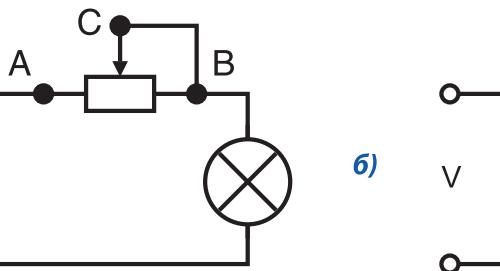
**Змінний резистор** призначений для регулювання сили струму або напруги в електричному колі. Має три виводи **A-B-C**. Опір між точками **A-B** постійний (у нашому випадку 5 кОм). При переміщенні повзунка (двигжка) опір між точками **A-C** і **B-C** змінюється. Коли повзунок знаходиться в точці **A**, то опір **A-C** дорівнює **0**, а опір **B-C** максимальний – 5 кОм. Коли повзунок у точці **B** – навпаки.

Опір вимірюється в омах (див. розділ «Резистори»).

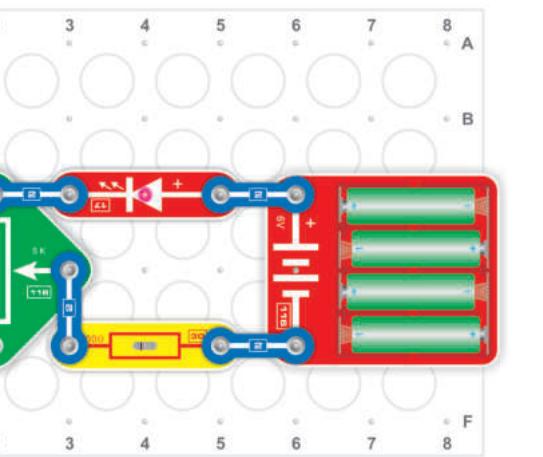
На електричних схемах змінні резистори можуть позначатися двома способами:



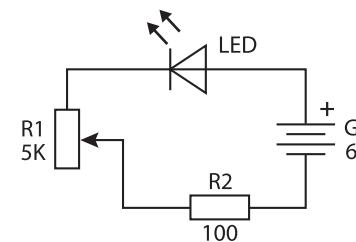
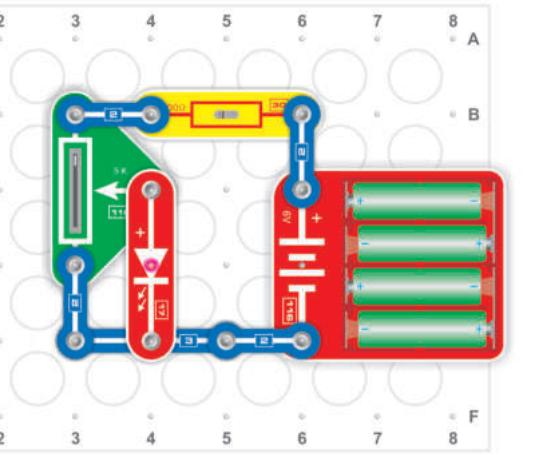
Змінний резистор має дві основні схеми увімкнення: **реостатну** (рис. а) та **потенціометричну** (рис. б). Як споживач зображена лампа.



## Реостатне увімкнення



## Потенціометричне увімкнення



Зберіть схеми. Змінюючи положення повзунка змінного резистора **118**, можна регулювати яскравість світіння світлодіода. У нижньому положенні повзунка опір максимальний і світлодіод світить дуже тъмяно. У міру переміщення повзунка вгору яскравість світлодіода буде збільшуватися. Можете замінити червоний світлодіод на зелений світлодіод, лампу **37** або електродвигун **38** та спробувати змінювати їх режим роботи за допомогою змінного резистора.

Змінні резистори бувають **повзункові**, як у нашому випадку, та **з коловою движковою системою**; бувають **однообертові**, а бувають **багатообертові**; бувають **підлаштовні** та **регулювальні**, **дротові** та **лакоплівочні**...

І кожен незамінний у своїй галузі.

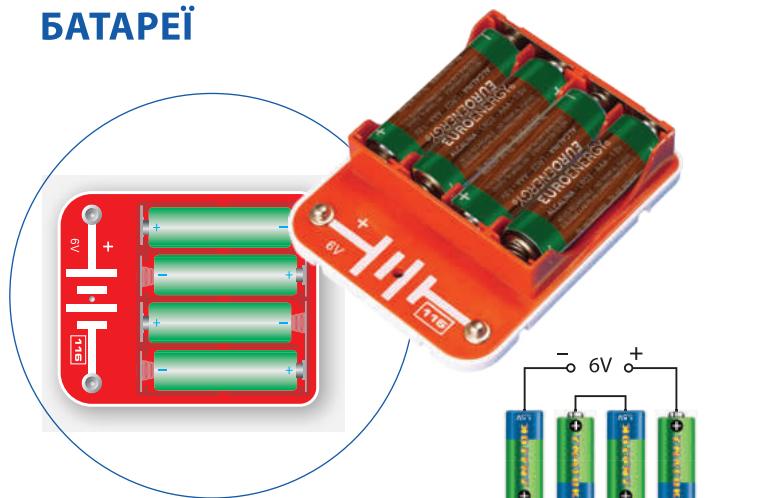


Світлодіоди з цього набору мають вбудовані в корпус захисні резистори, тому припустимо переміщення повзунка в крайнє верхнє положення (A), що неприпустимо для світлодіодів, які не мають захисного резистора.

## ДЖЕРЕЛА ЖИВЛЕННЯ

В цьому наборі використовуються два джерела живлення – блок батарей та USB-адаптер.

## БАТАРЕЇ

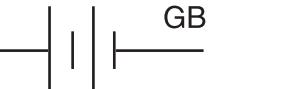


Батареї живлять схему. В даному наборі застосовані чотири батареї, підключенні послідовно. Максимальна напруга - 6В (4x1,5В), що набагато менше напруги, яка використовується в електричних мережах Вашого будинку. В даному наборі рекомендовано використовувати лужні батареї типу АА, 1,5В.

**Батареї купуйте окремо.**

**Батарейка – це зменшувально-пестливе від слова батарея.**

На електричних схемах батарейки позначаються ось так:



## USB-АДАПТЕР



З метою економії заряду батарейок можна живити схеми від мережі, використовуючи USB-адаптер, який складається з модуля 117, кабеля USB-miniUSB і блока живлення змінного струму (не входить до набору). Ви можете використовувати практично будь-який блок живлення змінного струму, за допомогою якого заряджаєте свій смартфон або планшет (вхід: ~220В/50Гц, вихід: 5В/0,5-2А).

Так виглядає USB-адаптер в зборі.

- **Дотримуйтесь полярності!**
- **Закривайте відсік для батарейок спеціальною кришкою, яка входить до комплекту.**
- **Не допускайте короткого замикання виводів модуля 117!**
- **Якщо якийсь з елементів схеми нагрівається, негайно відключіть джерело живлення від схеми та з'ясуйте причину.**
- **При нагріванні самих батарей, негайно відключіть їх від схеми та з'ясуйте причину. Наймовірніше, це помилка при збиранні схеми.**

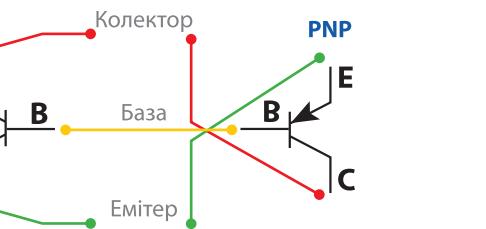
## ТРАНЗИСТОР



**Транзистор** – електронний напівпровідниковий пристрій, який в залежності від схеми включення призначений для посилення, перемикання, генерації або перетворювання електричних сигналів. Має три виводи – колектор **C**, база **B** та емітер **E**.

У цьому наборі використовуються кремнієві біполярні транзистори з NPN і PNP провідністю.

На електричних схемах транзистори позначаються ось таким чином:



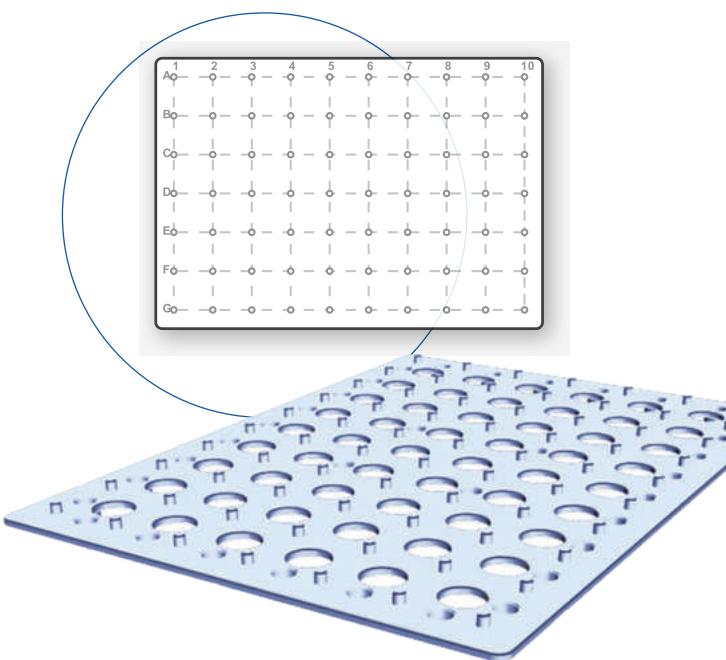
Якщо Ви бажаєте більше дізнатися про резистори, це можна зробити, придбавши Конструктор-ZNATOK-Школа (999+ схем) (подробиці на сайті <https://kiddisvit.ua/ua/brands/znatok>)

Під час роботи транзистор може сильно нагріватися і, щоб не обпектися, він закритий ковпачком. Без ковпачка транзистор виглядає ось так:



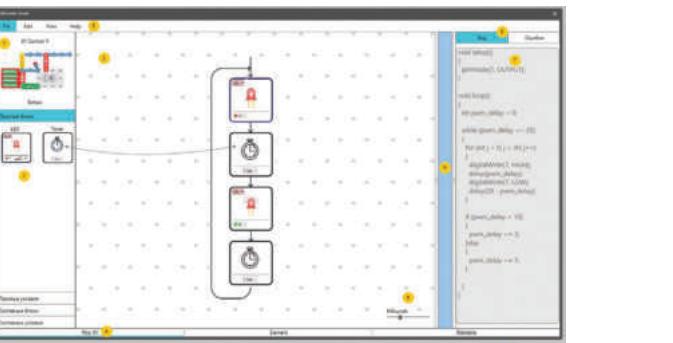
Монтажна плата – платформа для збирання на ній схем з деталей. На відміну від монтажних плат з інших наборів ZNATOK, тут є спеціальні кріплення з двох боків. На верхній частині знаходяться виступи для кріплення елементів, а на нижній частині – тримачі для мотор-редукторів та переходника для кріплення заднього колеса.

## МОНТАЖНА ПЛАТА





Найпростіший варіант, розроблений компанією «ZNATOK», для переходу від візуального програмування до програмування проектів вже «серйозно» мовою C++\Arduino. Склавши програму з простих блоків-дій, можна побачити, як ця ж програма виглядає мовою C++.

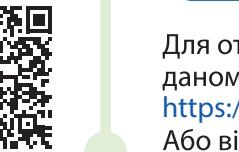


Виконавши ряд проектів, Ви зможете розуміти, як зміни в алгоритмі впливають на текст програми, та зможете вже самостійно писати код програм.

Для отримання інструкції для роботи з мовою Graph Z у даному конструкторі перейдіть за посиланням:

<https://bit.ly/3m9A329>

Або відскануйте цей QR-код:



Для програмування мовою C++ рекомендовано використовувати середовище Arduino IDE, яке містить бібліотеки для роботи з мікроконтролерами. Переважна частина програм, які написані для різних пристройів, була написана мовами C\C++, а використання модулів Arduino може бути першим серйозним кроком у вивченні програмування та мікроконтролерів.

```
01_Control_LED-10
1 #define BUTTON_PIN 7
2 #define LED_RED 5
3
4 void setup()
5 {
6     pinMode(LED_RED, OUTPUT);
7     pinMode(BUTTON_PIN, INPUT_PULLUP);
8 }
9
10 void loop()
11 {
12     if (digitalRead(BUTTON_PIN) == LOW)
13     {
14         digitalWrite(LED_RED, HIGH);
15     }
16     else
17     {
18         digitalWrite(LED_RED, LOW);
19     }
20 }
```

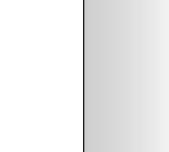
Усі програми, що зазначені в цьому керівництві і які Ви можете знайти на флешці, від самого початку написані саме для Arduino IDE, і в керівництві Ви зможете знайти приклади саме для цього варіанта.

**ПРИМІТКА:** Компанія «ZNATOK» не займається розробкою та підтримкою мови програмування C++ та середовища програмування Arduino IDE.

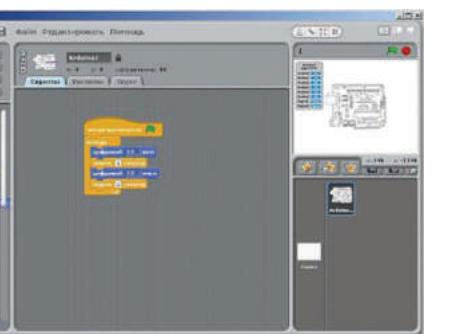
Для отримання інструкції для роботи з мовою C++ у даному конструкторі перейдіть за посиланням:

<https://bit.ly/3vEdlm8>

Або відскануйте цей QR-код:



Для візуального програмування можна використовувати графічну мову Scratch та її похідні платформи для робототехніки, наприклад, S4A (Scratch for Arduino) і mBlock. Даний варіант підіде для тих, хто тільки починає дізнатись, що таке алгоритми, умови, цикли – усі конструкції представлені наочно у вигляді блоків. Не всі проекти в даному конструкторі можуть бути запрограмовані за допомогою Scratch, однак ті, що входять до комплекту, дозволять розібратися в азах програмування та рухатись далі.



**ПРИМІТКА:** Компанія «ZNATOK» не займається розробкою та підтримкою мови програмування Scratch, а також платформ S4A і mBlock.

Для отримання інструкції для роботи з мовою Scratch в даному конструкторі перейдіть за посиланням: <https://bit.ly/2XHa4WY>

Або відскануйте цей QR-код:



Python не є основною мовою для програмування мікроконтролерів, але її все ж можна використовувати для керування пристроями. Якщо Ви хоча б трохи знаєте цю мову, то це може стати добром стартом для вивчення роботи мікроконтролерів. Не для всіх проектів з даного конструктора можна створити програму мовою Python, але Ви завжди можете створити щось своє!

```
while 1: #Do this forever
    var = raw_input() #get input from user
    print "you entered", var #print the input for confirmation

    if (var == '1'): #if the value is 1
        ArduinoSerial.write('1') #send 1
        print ("LED turned ON")
        time.sleep(1)

    if (var == '0'): #if the value is 0
        ArduinoSerial.write('0') #send 0
        print ("LED turned OFF")
```

Для отримання інструкції для роботи з мовою Python в даному конструкторі, перейдіть за посиланням: <https://bit.ly/3juGtY4>

Або відскануйте цей QR-код:



## ВСТАНОВЛЕННЯ ПРОГРАМ

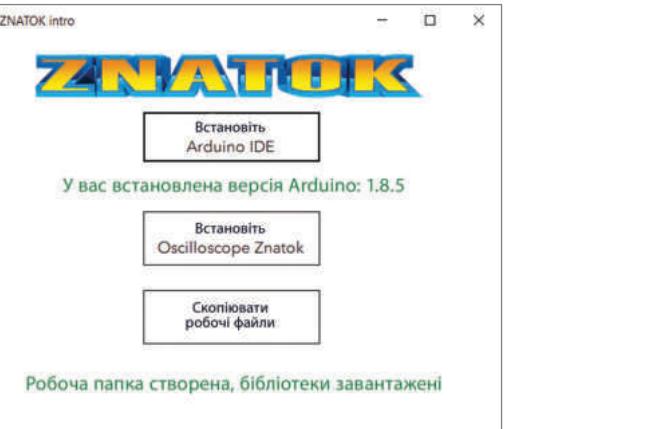
Ці програми, їхні оновлені версії, а також нові програми до цього конструктора, Ви можете знайти та безкоштовно скачати на сайті:  
<https://kiddisvit.ua/ua/brands/znatok/>

### Для користувачів Windows

1. Вставте флешку, що входить до комплекту, в USB-роз'єм Вашого комп'ютера.

2. Якщо у Вас автоматично не відкрилося вікно запуску, то відкрийте папку флешки вручну (**Цей комп'ютер-Znatok Arduino**) задопомогою Провідника Windows або іншого файлового менеджера (напр. Total Commander).

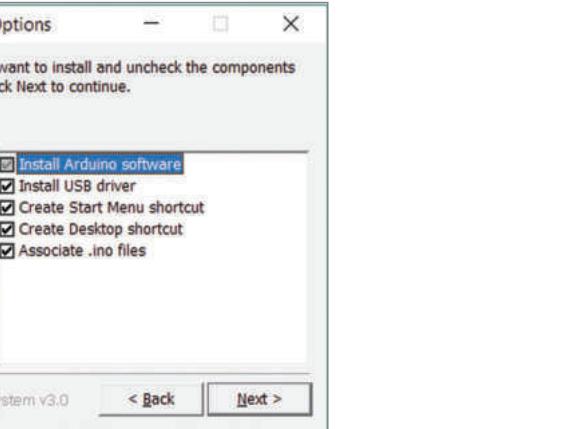
3. Запустіть програму **ZnatokIntro.exe**. Ця програма допоможе Вам встановити всі необхідні компоненти.



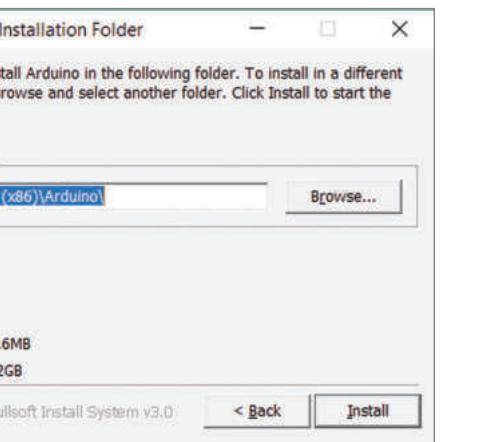
4. Для встановлення середовища програмування **Arduino IDE** натисніть кнопку **Встановити Arduino IDE**, якщо Arduino IDE потрібної версії вже була встановлена на Вашому комп'ютері, то цей крок не обов'язково робити.

4.1. Дотримуйтесь інструкцій інсталятора: на першому екрані, що Ви побачите, натисніть **I agree** («Я погоджуєсь»).

4.2. Обов'язково треба відмітити пункт **Install USB driver**. Натисніть **Next** («Далі»).



4.3. Оберіть шлях встановлення (рекомендовано обрати вже запропонований). Натисніть **Install** («Встановити»).



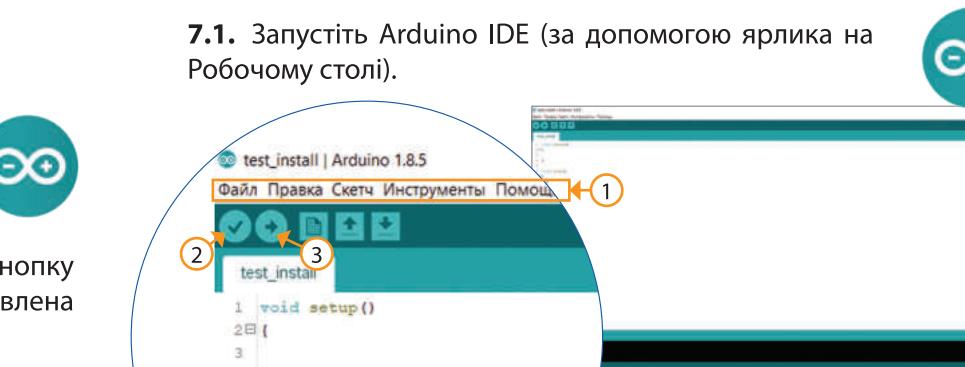
папка **Лабораторні роботи Znatok Arduino**, де будуть знаходитися усі файли програм.

Якщо Ви хочете вручну скопіювати файли програм, то можете знайти їх на флешці з даного набору, в папці **LaboratoryProjects**.

**Увага! Не рекомендовано відкривати файли програм безпосередньо з флешки!**

7. Підключіть до комп'ютера модуль **111** за допомогою USB-кабелю. Відбудеться встановлення обладнання. Якщо у Вас відключена функція автоматичного пошуку драйверів, то в ручному режимі їх можна знайти на флешці з даного набору, в папці **drivers**.

7.1. Запустіть Arduino IDE (за допомогою ярлика на Робочому столі).



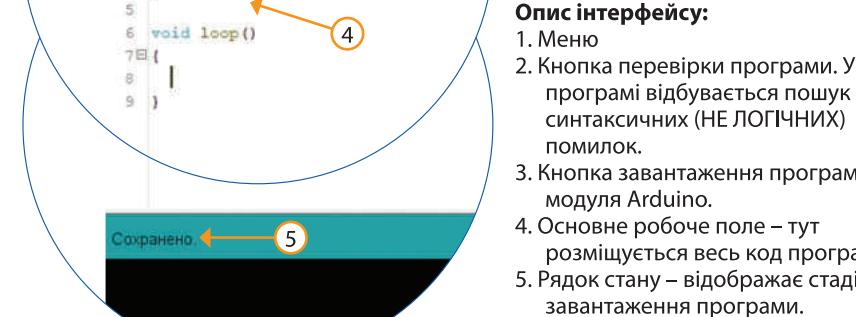
4.4. Дочекайтесь кінця встановлення, з'явиться надпис **Completed**, натисніть **Close** («Закрити»), після чого на Робочому столі з'явиться ярлик Arduino.

5. Для встановлення програми **Осцилограф** натисніть кнопку **Встановити ZnatokOscilloscope**, якщо ця програма була встановлена раніше, то цей крок необов'язково робити.

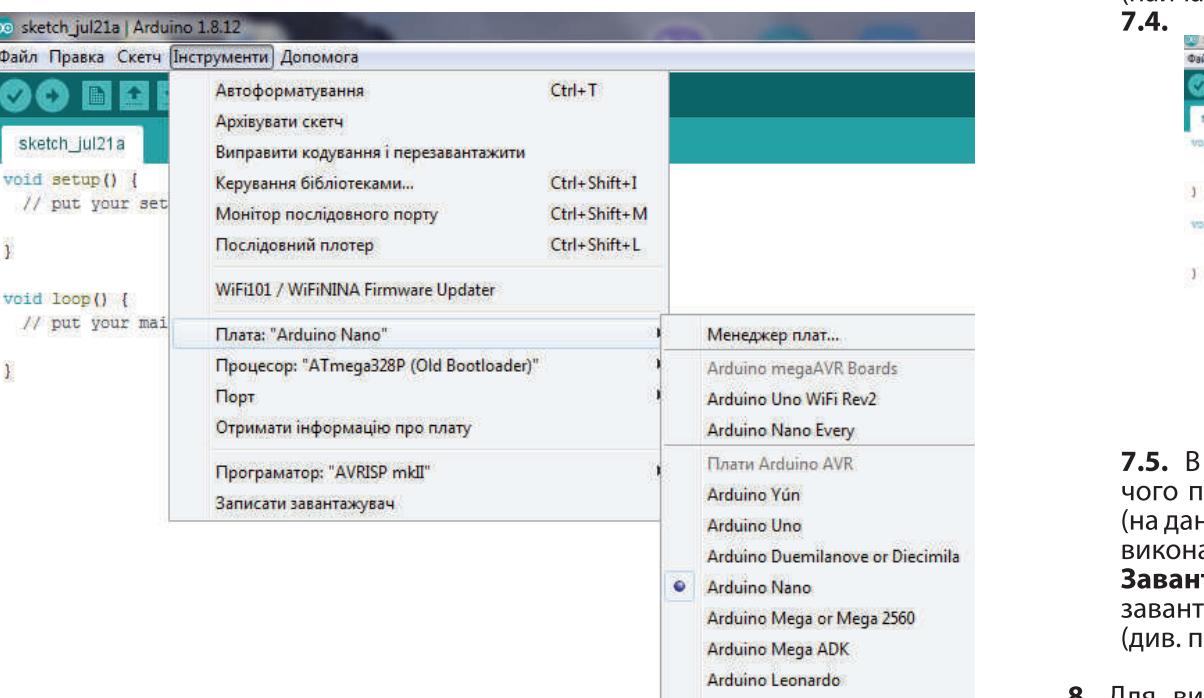
5.1. Дотримуйтесь інструкції інсталятора, натискаючи кнопку **Next** («Далі»)

5.2. Після успішного встановлення на Робочому столі з'явиться ярлик програми **Znatok Осцилограф**.

6. Для правильного виконання лабораторних робіт необхідно створити робочу папку на своєму комп'ютері, а також скопіювати необхідні файли – для цього натисніть кнопку **Скопіювати робочі файли**. Вам буде запропоновано місце створення папки, за замовчуванням – **Робочий стіл**. Натисніть кнопку **Обрати**, з'явиться

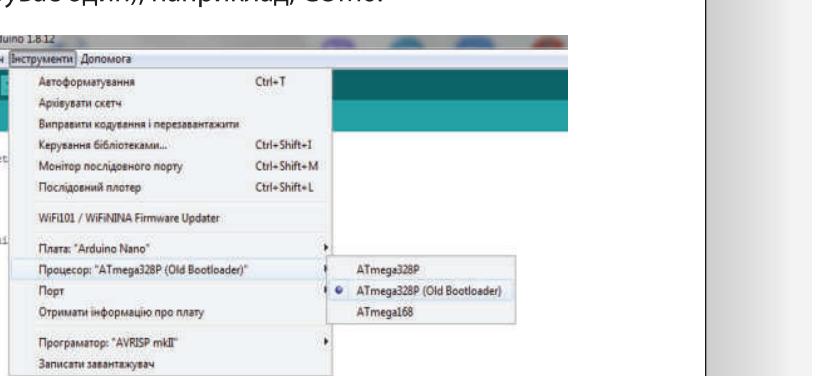


1. Меню
2. Кнопка перевірки програми. У програмі відбувається пошук синтаксичних (НЕ ЛОГІЧНИХ) помилок.
3. Кнопка завантаження програми до модуля Arduino.
4. Основне робоче поле – тут розміщується весь код програми.
5. Рядок стану – відображає стадію завантаження програми.



**7.2.** У меню **Інструменти** оберіть підпункт **Плата** і у випадаючому списку оберіть **Arduino Nano**.

**7.3.** В меню **Інструменти** оберіть підпункт **Порт** і у випадаючому списку оберіть будь-який підпункт (найчастіше буває один), наприклад, COM6.



**7.5.** В меню **Скетч** оберіть підпункт **Завантаження**, після чого почнеться завантаження програми до модуля **111** (на даний момент програма тестова), якщо усі кроки були виконані вірно, то у рядку стану з'явиться повідомлення **Завантаження завершено**. Якщо виникла проблема завантаження до модуля **111**, то спробуйте змінити порт (див. п. 7.3.).

**8.** Для виконання лабораторної роботи відкрийте папку **Лабораторні роботи Znatok Arduino** на Вашому комп'ютері. Оберіть потрібну лабораторну роботу та відкрийте програму – відкриється Arduino IDE з текстом програми. Якщо Arduino IDE вже відкрита, то Ви можете в меню **Файл** обрати підпункт **Відкрити** та обрати потрібну програму.

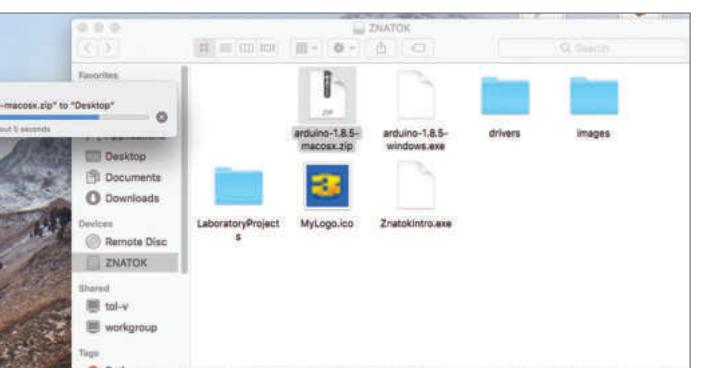
**8.1.** Для завантаження програми в Arduino необхідно обрати в меню **Скетч** підпункт **Завантаження** або клікнути по відповідній піктограмі. **НЕ ЗАБУВАЙТЕ:** при помилці завантаження перевірте правильність підключення модуля **111** до Вашого комп'ютера через USB-кабель, а також правильність вибору порту (див. п. 7.3.).

## Для користувачів macOS

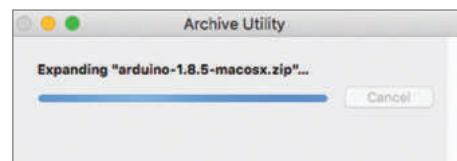
**1.** Вставте флешку, що входить до комплекту, в USB-роз'єм Вашого комп'ютера, на робочому столі з'явиться ярлик флешки:



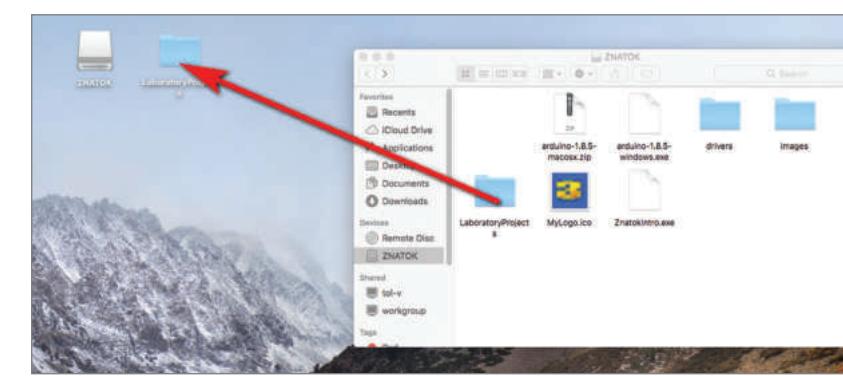
**2.** Скопіюйте архів **arduino-1.8.12-macosx.zip**, наприклад, на Робочий стіл.



**3.** Розпакуйте файл **arduino-1.8.12-macosx.zip** у потрібне Вам місце (рекомендовано «Application»), після чого програма буде доступна для запуску.



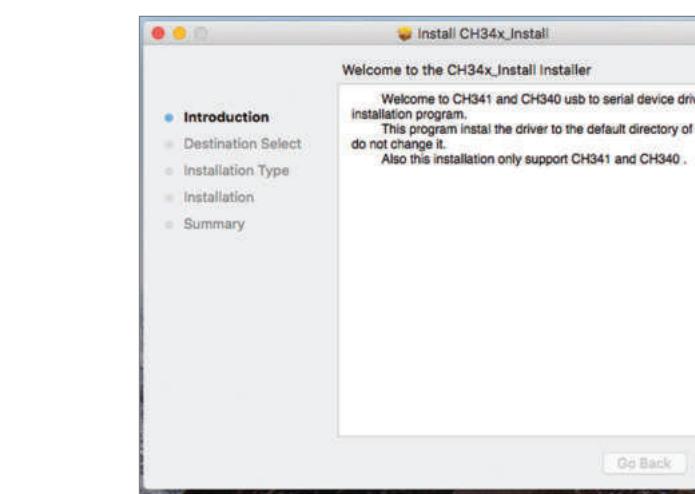
**4.** Скопіюйте папку з лабораторними роботами **LaboratoryProjects** у будь-яке зручне для Вас місце.



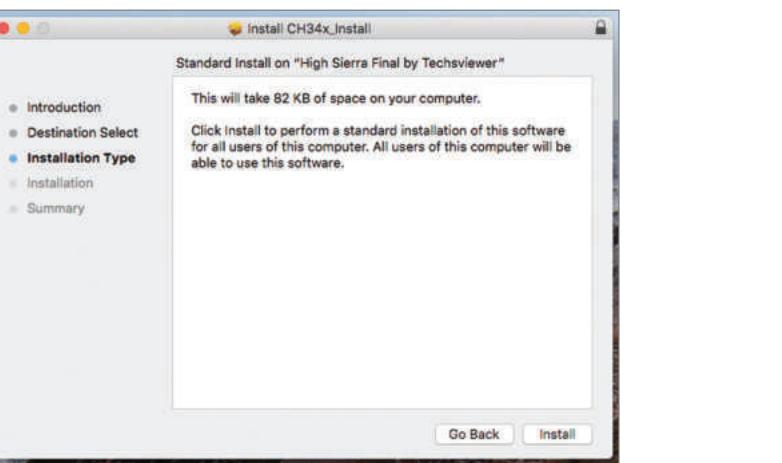
**5.** В папці **drivers\CH341SER\_MAC**, на флешці, що ВХОДИТЬ ДО КОМПЛЕКТУ, відкрийте файл інсталятора:



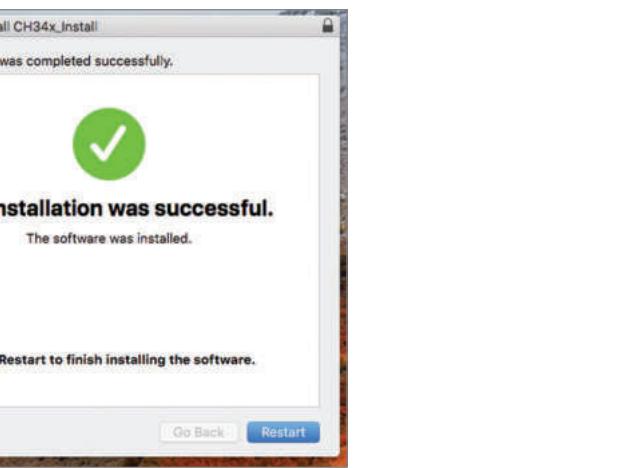
**6.** Дотримуйтесь інструкцій інсталятора:



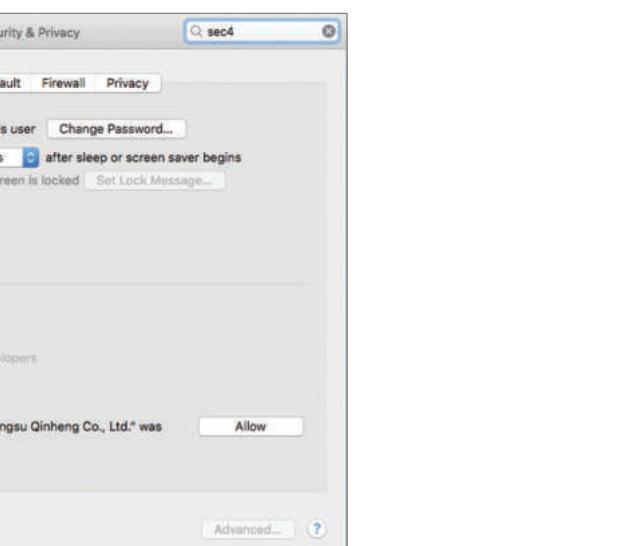
Під час встановлення треба буде ввести пароль користувача системи:



Під час встановлення треба буде ввести пароль користувача системи:

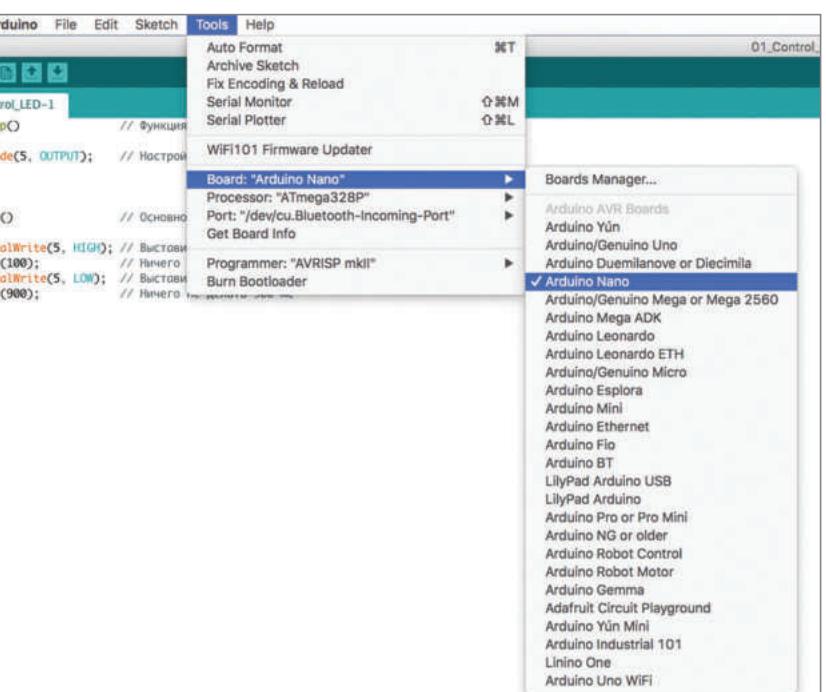


7. В налаштуваннях **Security&Privacy** Вашого комп'ютера, необхідно натиснути кнопку **Allow** для коректної роботи драйверів.

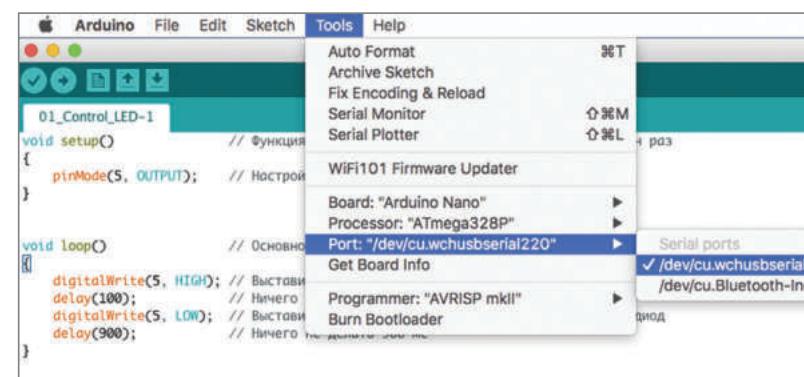


8. Запустіть **Arduino IDE** (з папки **Applications**, якщо Ви встановили програму в цю папку).

8.1. В меню **Tools** оберіть підпункт **Board** та у випадаючому списку оберіть **Arduino Nano**.



8.2. В меню **Tools** оберіть підпункт **Port** та у випадаючому списку оберіть підпункт, який зображений на рисунку.



8.3. В меню **Sketch** оберіть підпункт **Upload**, після чого почнеться завантаження програми до модуля **111** (на даний момент програма тестова), якщо усі кроки були виконані правильно, то у рядку стану з'явиться повідомлення «**Завантаження завершено**». Якщо виникла проблема завантаження до модуля **111**, то спробуйте змінити Порт (**див. п. 8.2.**).

9. Для виконання лабораторної роботи, відкрийте папку **Лабораторні роботи Znatok Arduino** на Вашому комп'ютері. Оберіть потрібну лабораторну роботу та відкрийте програму – відкриється **Arduino IDE** з текстом програми.

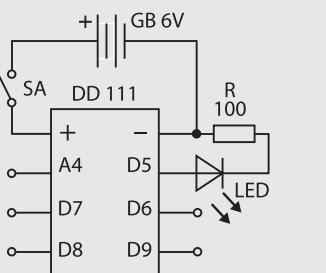
Якщо Arduino IDE вже відкрита, то Ви можете в меню **File** обрати підпункт **Open** та обрати потрібну програму.

9.1. Для завантаження програми в Arduino необхідно обрати в меню **Sketch** підпункт **Upload** або клікнути по відповідній піктограмі. **НЕ ЗАБУВАЙТЕ:** при помилці завантаження перевірте правильність підключення модуля **111** до Вашого комп'ютера через USB-кабель, а також правильність вибору порту (**див. п. 8.2.**).

## КЕРУВАННЯ СВІТЛОДІОДАМИ



**ZNATOK™**

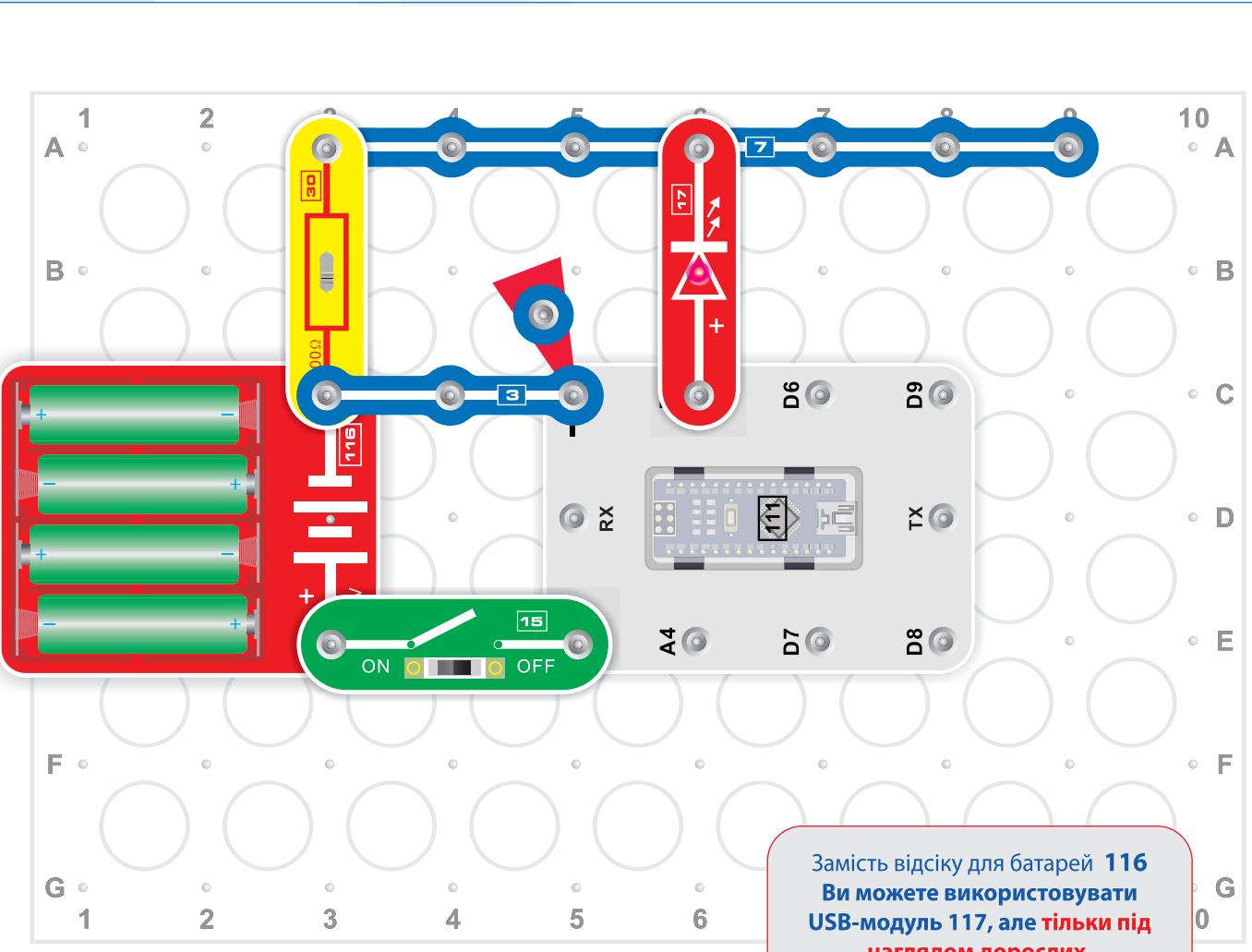


Зберіть коло, схему якої зображено на рисунку. До цієї схеми додаються 5 програм. Підключіть модуль 111 до комп'ютера за допомогою USB-кабелю та завантажте програму 01\_Control\_LED-1. Після завантаження програми\* від'єднайте кабель та увімкніть живлення – вимикач 15 в положенні ON.

Світлодіод повинен періодично починати світити та згасати.

Далі наведені програми, у яких вказані параметри, які можна налаштовувати для зміни роботи схеми. Програми можна також переглянути на флешці, що входить до комплекту.

\*після завантаження зелений та червоний світлодіоди на модулі 111, припиняють миготіти, і низу екрана комп'ютера в рядку стану з'явиться надпис «Завантаження завершено».



## ПРОГРАМИ

### 01\_Control\_LED-1

Дана програма дозволяє світлодіоду працювати в режимі «Спалах» – вимкає його на короткий час. Змінюючи числові значення в програмі, можна регулювати тривалість світіння світлодіода. Нижче наведений повний текст програми з

```
delay(100)
Збільшіть це
значення до 500 і
переконайтесь, що
світлодіод став
горіти довше

void setup()
// Функція встановлення - команди всередині виконуються один раз
pinMode(5, OUTPUT);
// Налаштування цифрового виводу D5 на вхід

}

void loop()
{
delay(900)
Зменшіть це
значення до 100
і переконайтесь,
що світлодіод став
гаснути на менший
час

// Виставити «високий» рівень на D5 - увімкнути світлодіод
digitalWrite(5, HIGH);
delay(100);
// Нічого не робити 100 мс
// Виставити «низький» рівень на D5 - вимкнути світлодіод
digitalWrite(5, LOW);
delay(900);
// Нічого не робити 900 мс
}
```

### 01\_Control\_LED-2

Вимкніть живлення схеми, під'єднайте USB-кабель, завантажте програму, від'єднайте кабель, увімкніть живлення.

Дана програма дозволяє світлодіоду працювати в режимі «Маячок» – вимикає його на короткий час.

коментарями та виділені параметри, які змінюють режим роботи.

Праворуч наведена блок-схема роботи цієї програми.



Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля 111(Arduino)!

# АМИ

## 01\_Control\_LED-3

Вимкніть живлення схеми, під'єднайте USB-кабель, завантажте програму, від'єднайте кабель, увімкніть живлення.

Дана програма поступово збільшує яскравість світлодіода до максимальної. Замінюючи параметри в тексті програми, можна змінити мінімальне та максимальне значення яскравості світлодіода, текст пр  
режим р  
Правс

те, наскільки плавно світлодіод буде починати світити і як уде виконуватися увесь цикл. Нижче наведений повний алгоритм з коментарями до виділених параметрів, які змінюють

ограмми з коментарями та виділені параметри, які змінюють роботи.

Задача 2. Відповідь на цю питання можна отримати, якщо звернутися до блок-схеми роботи цієї програми.

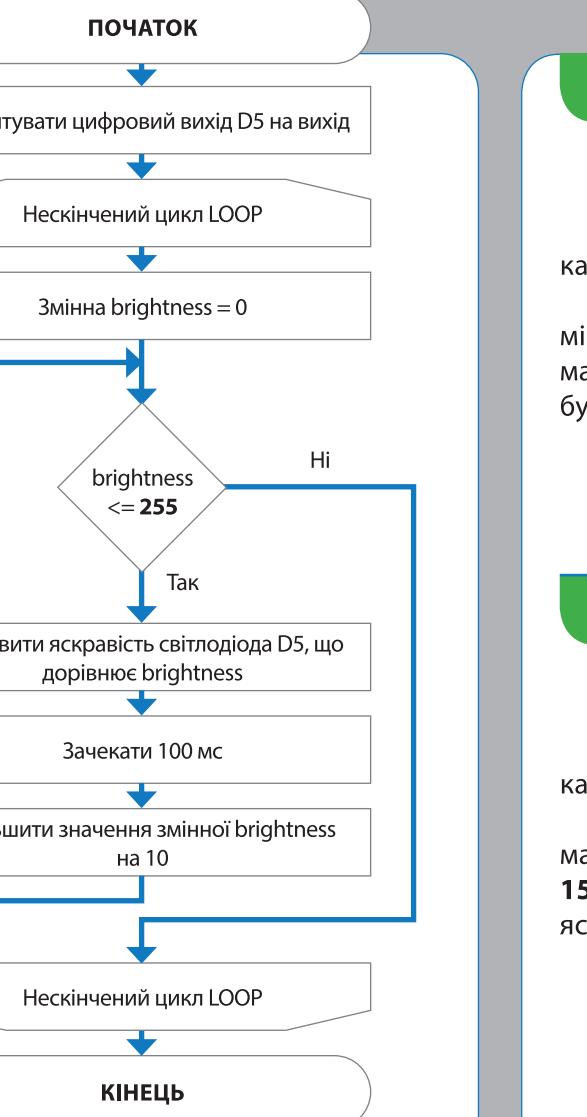
```
void setup()
{
    pinMode(5, OUTPUT);
}

void loop()
{
    int brightness = 0;
    while (brightness <= 255)
    {
        analogWrite(5, brightness);
        delay(100);
        brightness = brightness + 10;
    }
}
```

**Мінімальне значення яскравості**  
Збільшіть це значення до 50 і переконайтесь, що світлодіод ніколи повністю не згасає

**Максимальне значення яскравості**  
Зменшіть це значення до 100 і переконайтесь, що світлодіод ніколи не засвітиться на повну силу

0



## 01\_Control\_LED-4

Вимкніть живлення схеми, під'єднайте USB-кабель, завантажте програму, від'єднайте кабель, увімкніть живлення.

Дана програма поступово зменшує яскравість світлодіода від максимальної мінімальної. Змінюючи параметри в тексті програми, можна змінити мінімальну максимальне значення яскравості світлодіода, а також те, наскільки плавно світло буде починати світити і як довго буде виконуватися увесь цикл.

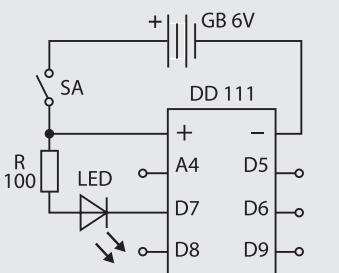
## 01\_Control\_LED-5

Вимкніть живлення схеми, під'єднайте USB-кабель, завантажте програму, від'єднайте кабель, увімкніть живлення.

Дана програма об'єднує **01\_Control\_LED-3** і **01\_Control\_LED-4**, але за великою максимальної яскравості взято число **150** — це зроблено тому, що зміна яскравості **150** до **255** не помічається оком і здається, що світлодіод просто світить з максимальною яскравістю.

ної зміні в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забудьте завантажувати оновлену програму до модуля **111**(Arduino)!

## КЕРУВАННЯ СВІТЛОСХІДОДАМИ (ПРОДОВЖЕННЯ)



Зберіть схему, як зображенено на рисунку. До цієї схеми додається 1 програма. Підключіть модуль 111 до комп'ютера за допомогою USB-кабелю та завантажте програму **01\_Control\_LED-6**. Після завантаження програми\* від'єднайте кабель та увімкніть живлення – вимикач 15 в положенні ON.

Яскравість світіння світлодіода почне поступово змінюватися. При чому, швидкість зміни яскравості на початку роботи буде швидшою, ніж наприкінці.

Далі наведена програма, у якій вказані параметри, які можна налаштовувати для зміни роботи схеми. Програму можна також переглянути на флешці, що входить до комплекту.

\*після завершення завантаження зелений та червоний світлодіоди на модулі 111, припиняють миготіти і внизу екрана комп'ютера в рядку стану з'явиться напис «Завантаження завершено».



**ZNATOK™**

Це відео допоможе зрозуміти, як він працює

Яскравість LED

Вихід D7

max

min

t

T X1 T X2 T X3 T X4 T

01\_Control\_LED-6

1 2 3 A B C D E F G

D5 D6 D9 RX TX A4 D8

15 ON OFF

5 6 7 8 9 10

116

**ZNATOK™**

## ПРОГРАМИ

### 01\_Control\_LED-6

Після завантаження, програма дозволить реалізувати **ШІМ** (див. стор. 15) на виводі D7, який не призначений для роботи з функцією `analogWrite()`. Це дає можливість не тільки вимикати та вимикати світлодіод, але й змінювати яскравість його світіння.

В цій схемі можлива зміна тривалості періоду ШІМ, кількості цих періодів, а також швидкості змінення яскравості, яка визначається різницею між величинами X1 і X2, X2 і X3 (див. графік на попередній сторінці). Нижче наведена програма та виділені параметри, які змінюють режим роботи.

**Зміна тривалості циклу**  
Збільшіть дане значення з 30 до 90, тим самим збільшивши час, протягом якого світлодіод буде горіти на кожному рівні яскравості.

**Швидкість зміни яскравості**  
Поміняйте дані значення місцями і переконайтесь, що світлодіод став на початку загоратися повільніше.

```
while (pwm_delay <= 20)
{
    for (int j = 0; j < 30; j++)
    {
        digitalWrite(7, HIGH);
        delay(pwm_delay);
        digitalWrite(7, LOW);
        delay(20 - pwm_delay);
    }
    if (pwm_delay < 10)
    {
        pwm_delay += 3;
    }
    else
    {
        pwm_delay += 1;
    }
}
```

Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля 111(Arduino)!



## ПРОГРАМИ

## 01\_Control\_LED-8

Після завантаження, програма буде працювати в режимі «Вогні, що біжать» та виконувати наступну послідовність дій: три світлодіоди по черзі почнуть світити з заданим інтервалом часу. Потім цикл повториться.

В цій схемі можлива зміна швидкості роботи та послідовності увімкнення. Нижче наведений фрагмент з програми та виділені параметри, які змінюють режим роботи.

```
...
void loop()
{
    digitalWrite(LED_RED, HIGH);
    delay(200);
    digitalWrite(LED_RED, LOW);

    digitalWrite(LED_YELLOW, HIGH);
    delay(200);
    digitalWrite(LED_YELLOW, LOW);

    digitalWrite(LED_GREEN, HIGH);
    delay(200);
    digitalWrite(LED_GREEN, LOW);
}
...
```

**Швидкість зміни**

Зменшіть дані значення, наприклад, до 5, а потім збільшіть до 500. Опишіть різницю у вигляді схеми.

## 01\_Control\_LED-9

Після завантаження, програма буде працювати в режимі «Вогні, що біжать з прискоренням» та виконувати наступну послідовність дій: три світлодіоди по черзі почнуть світити спочатку з одним інтервалом часу, потім з меншим інтервалом, потім ще з меншим, доки інтервал між увімкненням не стане мінімальним. Потім цикл повториться.

В цій схемі можлива зміна швидкості роботи та послідовності увімкнення. Нижче наведений фрагмент з програми та виділені параметри, які змінюють режим роботи.

```
...
int switch_delay = 1000;
...
if (switch_delay > 200)
{
    switch_delay -= 200;
}
else
{
    switch_delay -= 5;
}
...
```

**Початкова швидкість**

Зменшіть це значення до 300 та переконайтесь, що з самого початку вогні будуть бігти значно швидше.

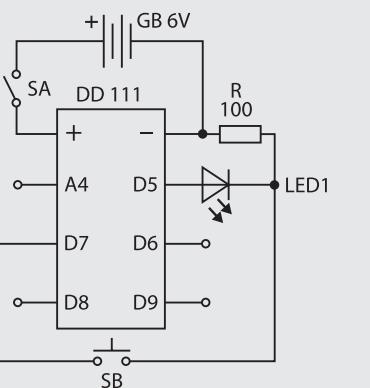
**Початковий час затримки**

Зменшіть це значення до 50 та переконайтесь, що спочатку – на невеликій швидкості перемикання – світлодіоди стали працювати довше.

## КЕРУВАННЯ СВІТЛОСДОДАМИ (ПРОДОВЖЕННЯ)



**ZNATOK™**



Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додаються 5 програм:

### 01\_Control\_LED-10

«Проста кнопка»

### 01\_Control\_LED-11

«Тривале світіння»

### 01\_Control\_LED-12

«Миготіння»

### 01\_Control\_LED-13

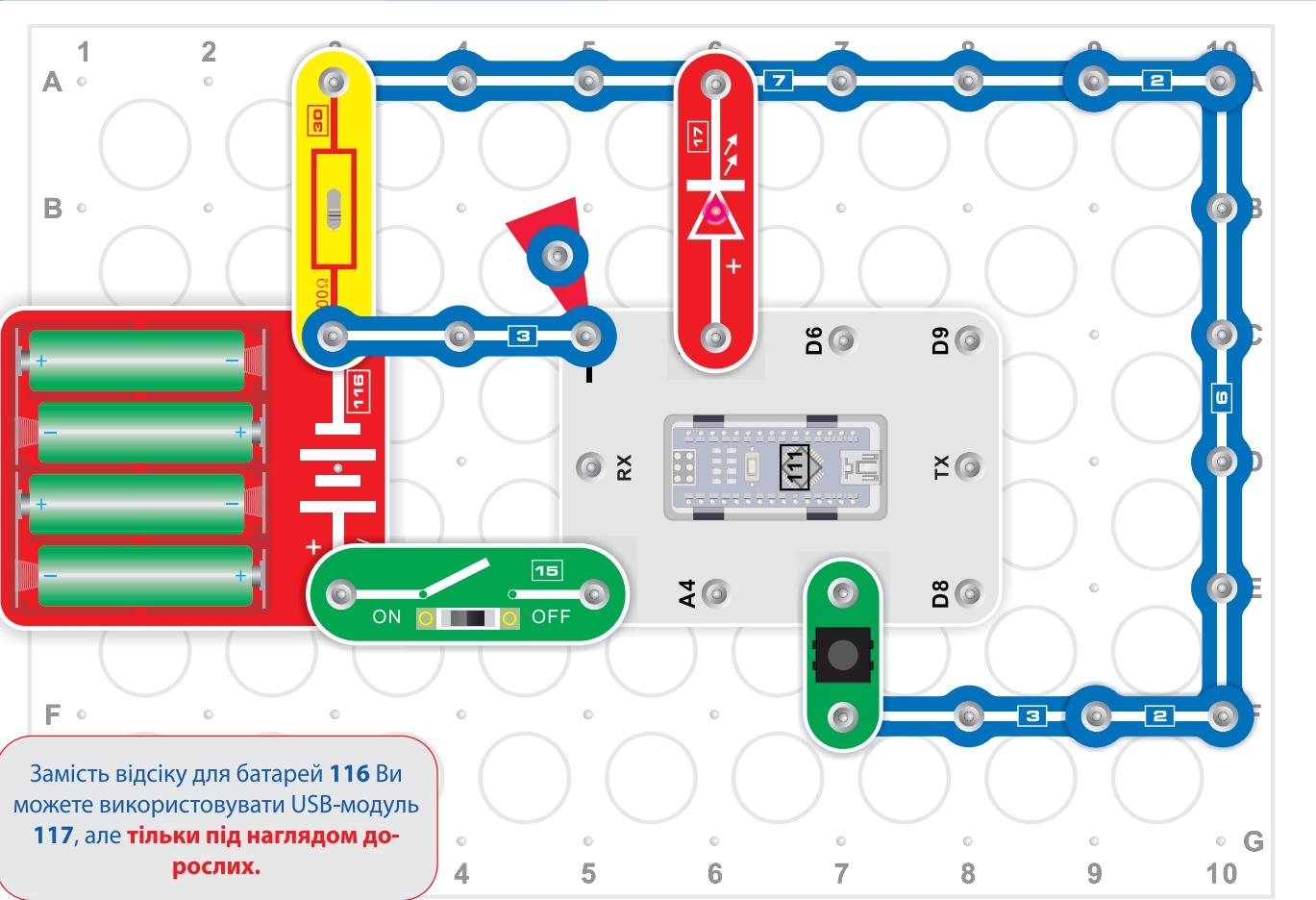
«Плавна яскравість»

### 01\_Control\_LED-14

«Кнопка з фіксацією»

Ці програми дозволяють по-новому подивитися на можливості звичайної кнопки, коли вона підключена до мікроконтролера.

Підключіть модуль **111** до комп'ютера задопомогою USB-кабелю та завантажте одну з наведених програм. Після завантаження



програми від'єднайте кабель та увімкніть живлення – вимикач **15** у положенні ON.

Далі наведені описи програм і тих фрагментів, де вказані параметри, які можна міняти для зміни роботи схеми. Програми можна також переглянути на флешці, що входить до комплекту.

**ZNATOK™**

## ПРОГРАМИ

### 01\_Control\_LED-10

Після завантаження програми схема буде працювати таким чином: світлодіод буде світити тільки тоді, коли кнопка натиснута. Це найпростіший режим та найпростіша програма.

```
...
void loop()
{
    if (digitalRead(BUTTON_PIN) == LOW) // Якщо рівень напруги на виводі BUTTON_PIN низький
    {
        digitalWrite(LED_RED, HIGH);
    }
    else // Інакше
    {
        digitalWrite(LED_RED, LOW);
    }
...
}
```

**Інверсія**  
Змініть даний параметр на HIGH і переконайтесь, що тепер, коли кнопка натиснена, світлодіод не світить.

### 01\_Control\_LED-11

```
...
const int blinkDelay = 3000; // Час горіння
...

```

**Час світіння**  
Змініть даний параметр на 1000 і переконайтесь, що тепер світлодіод світить набагато менше часу. Потім збільшіть це значення до 6000.

Після кожної зміни в програмі зберігайте оновлену версію в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля **111**(Arduino)!

## ПРОГРАМИ

## 01\_Control\_LED-12

Після завантаження програми схема буде працювати таким чином: при короткочасному натисканні на кнопку світлодіод блимне 5 разів. Потім вимкнеться. Ця функція корисна при організації «кнопки тривоги» - одне натискання запускає серію спалахів, яку неможливо зупинити, якщо не відключити живлення.

В цій схемі можлива зміна кількості увімкнень світлодіода після того, як відпустити кнопку, та зміна частоти цих увімкнень. Нижче наведені два фрагменти з програми та виділені параметри, які змінюють режим роботи.

```
...
const int debounceTime = 5; // Час захисту від брязку
const int blinkDelay = 300; // Частота включення
boolean lastButtonState = false; // Попередній стан кнопки
...

void blinking()
{
    for (int i = 0; i < 5; i++)
    {
        digitalWrite(LED_RED, HIGH);
        delay(blinkDelay);
        digitalWrite(LED_RED, LOW);
        delay(blinkDelay);
    }
}
...
```

**Частота увімкнень**

Зменшіть цей параметр на 1000 та переконайтесь, що тепер світлодіод почав миготіти рідше.

**Кількість увімкнень**

Змініть цей параметр на 3 та переконайтесь, що світлодіод почав миготіти 3, а не 5 разів.

## 01\_Control\_LED-13

Після завантаження програми схема буде працювати таким чином: при натисканні та утриманні кнопки яскравість світіння світлодіода буде поступово збільшуватися до максимальної. Після повторного натискання на кнопку, світлодіод плавно вимкнеться. Ця функція корисна при організації освітлення – Ви самі встановлюєте потрібний рівень яскравості.

У цій схемі можлива зміна плавності увімкнення світлодіода та зміна його максимальної яскравості. Нижче наведений фрагмент з програми та виділені параметри, які змінюють режим роботи.

```
...
void switching()
{
    int max_brightness = 255;
    if (isLedOn)
    {
        for (int i = 0; i < max_brightness; i+=max_brightness-1)
        {
            analogWrite(LED_RED, i);
            delay(20);
        }
        isLedOn = true;
    }
    else
    {
        for (int i = max_brightness; i > 0; i-=max_brightness)
        {
            analogWrite(LED_RED, i);
            delay(20);
        }
        digitalWrite(LED_RED, LOW);
        isLedOn = false;
    }
}
...
```

**Плавність увімкнень**

Змініть цей параметр на 5 та переконайтесь, що тепер світлодіод почав збільшувати свою яскравість значно швидше

**Максимальна яскравість**

Змініть цей параметр на 50 та переконайтесь, що при досягненні максимальної яскравості світлодіод почав світити менш яскраво

Після завантаження програми схема буде працювати таким чином: при короткочасному натисканні на кнопку світлодіод почне світити і буде продовжувати світити до тих пір, поки на кнопку не натиснути ще раз. Але, на відміну від звичайного вимикача, ми можемо подбати про очі та зробити так, щоб світло вмикалося плавно, не осліплюючи.

У цій схемі можлива зміна плавності увімкнення світлодіода та зміна його максимальної яскравості. Нижче наведений фрагмент з програми та виділені параметри, які змінюють режим роботи.

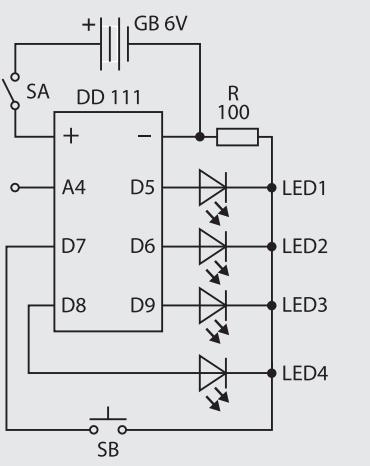
```
...
void switching()
{
    int max_brightness = 255;
    if (isLedOn)
    {
        for (int i = 0; i < max_brightness; i+=max_brightness-1)
        {
            analogWrite(LED_RED, i);
            delay(20);
        }
        isLedOn = true;
    }
    else
    {
        for (int i = max_brightness; i > 0; i-=max_brightness)
        {
            analogWrite(LED_RED, i);
            delay(20);
        }
        digitalWrite(LED_RED, LOW);
        isLedOn = false;
    }
}
...
```

**Плавність увімкнення**

Змініть «`max brightness-1`» (максимальну яскравість) на 1 та переконайтесь, що світлодіод почав умикатися плавно.

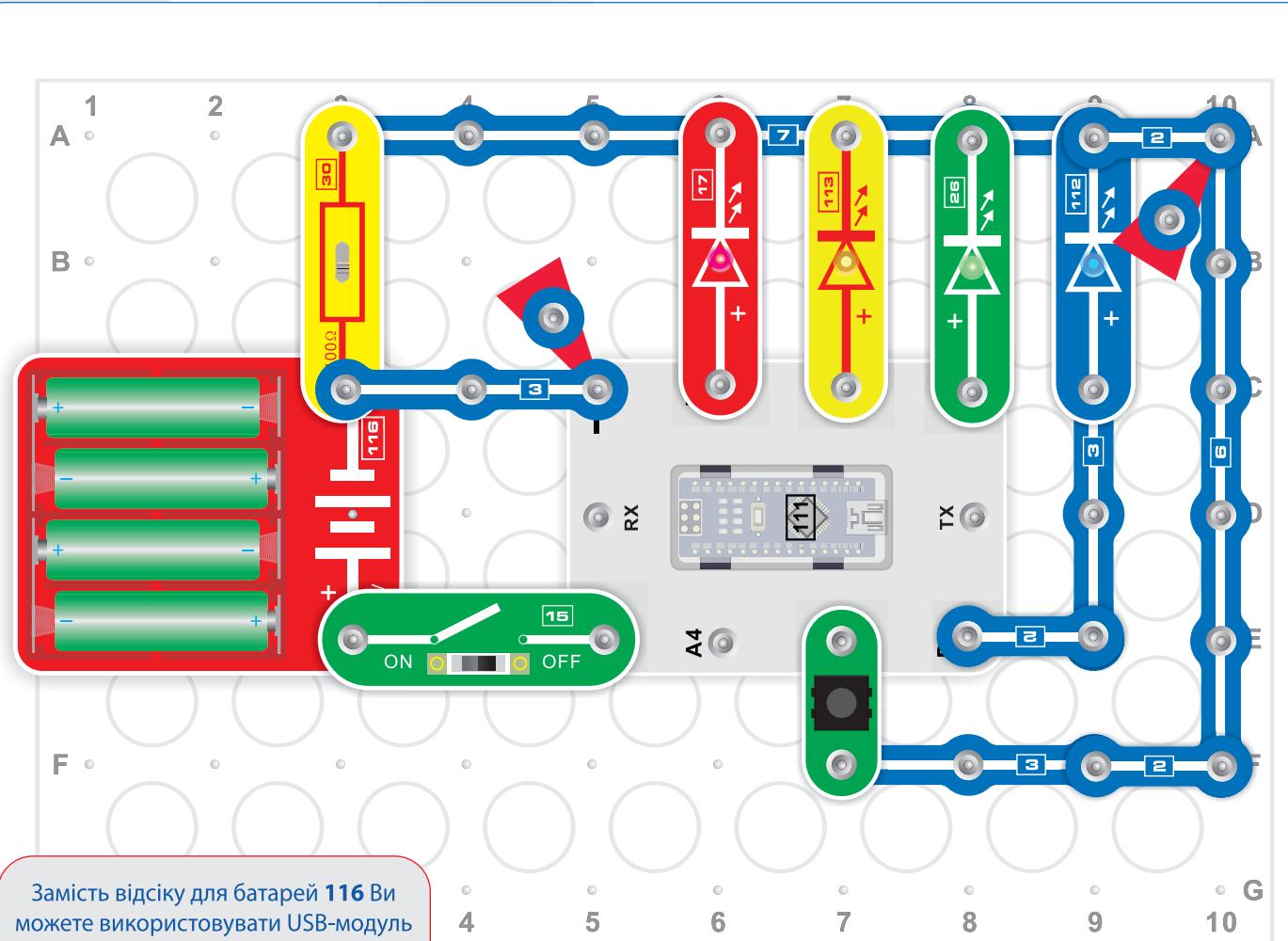
**Плавність вимкнення**

Змініть «`max brightness-1`» (максимальну яскравість) на 10 та переконайтесь, що світлодіод почав вимикатися плавно.



Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 1 програма: **01\_Control\_LED-15**. Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте одну з наведених програм. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач **15** в положенні **ON**.

Через те що програма об'ємна, вона не наведена в опису цього керівництва, але подивитися текст програми з коментарями можна на флешці, що входить до комплекту.



## ПРОГРАМИ

## 01\_Control\_LED-15

Після завантаження програми, схема почне працювати в режимі «Вогні, що біжать - зліва направо». Кожне натискання на кнопку буде послідовно запускати наступні режими:

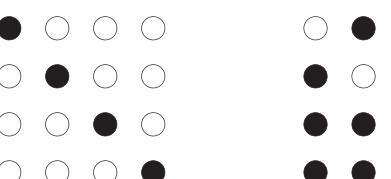
- «Вогні, що біжать - зліва направо»
- «Вогні, що біжать з прискоренням - зліва направо»
- «Вогні, що біжать – справа наліво»
- «Складання»
- «Тінь, що біжить»
- «Попарне увімкнення 1»
- «Попарне увімкнення 2»
- «Бігунок»

Потім цикл повторюється.

Схожі режими роботи використовуються в сучасних гірляндах для ялинок та на рекламних табло. Збільшуючи кількість світлодіодів, та змінюючи їх розміщення, можна досягти приголомшливих світлових ефектів.

## Діаграми роботи:

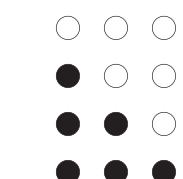
## «Вогні, що біжать»



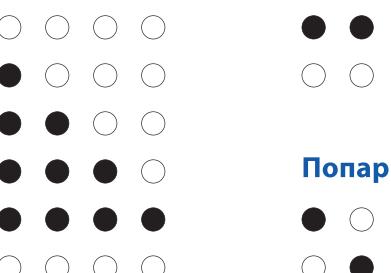
## «Тінь, що біжить»



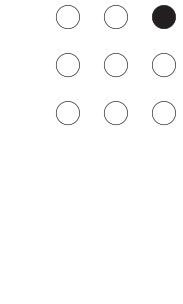
## «Бігунок»



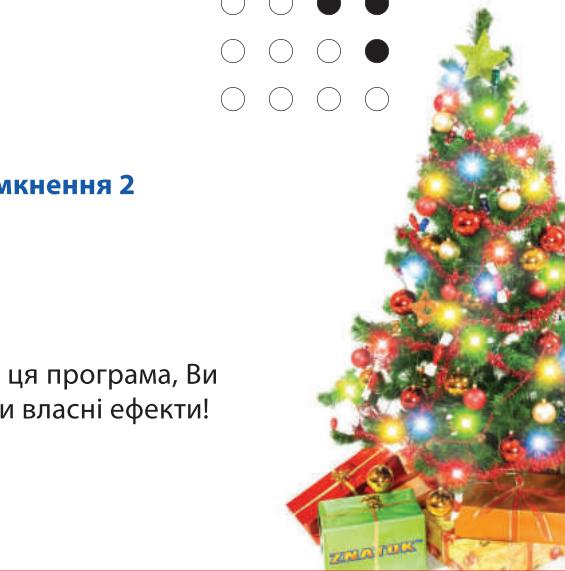
## «Складання»



## Попарне увімкнення 1



## Попарне увімкнення 2



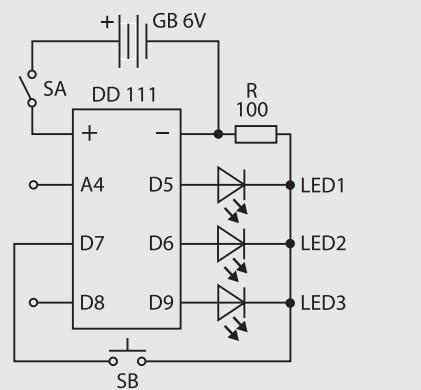
Розібравшись у тому, як працює ця програма, Ви можете внести зміни та створювати власні ефекти!

Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля **111**(Arduino)!

## КЕРУВАННЯ СВІТЛОСВІДОДАЧАМИ (ПРОДОВЖЕННЯ)



**ZNATOK™**



Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додаються 3 програми:

**01\_Control\_LED-17** —

«Послідовний вибір»

**01\_Control\_LED-18** —

«Довільний вибір»

**01\_Control\_LED-19** —

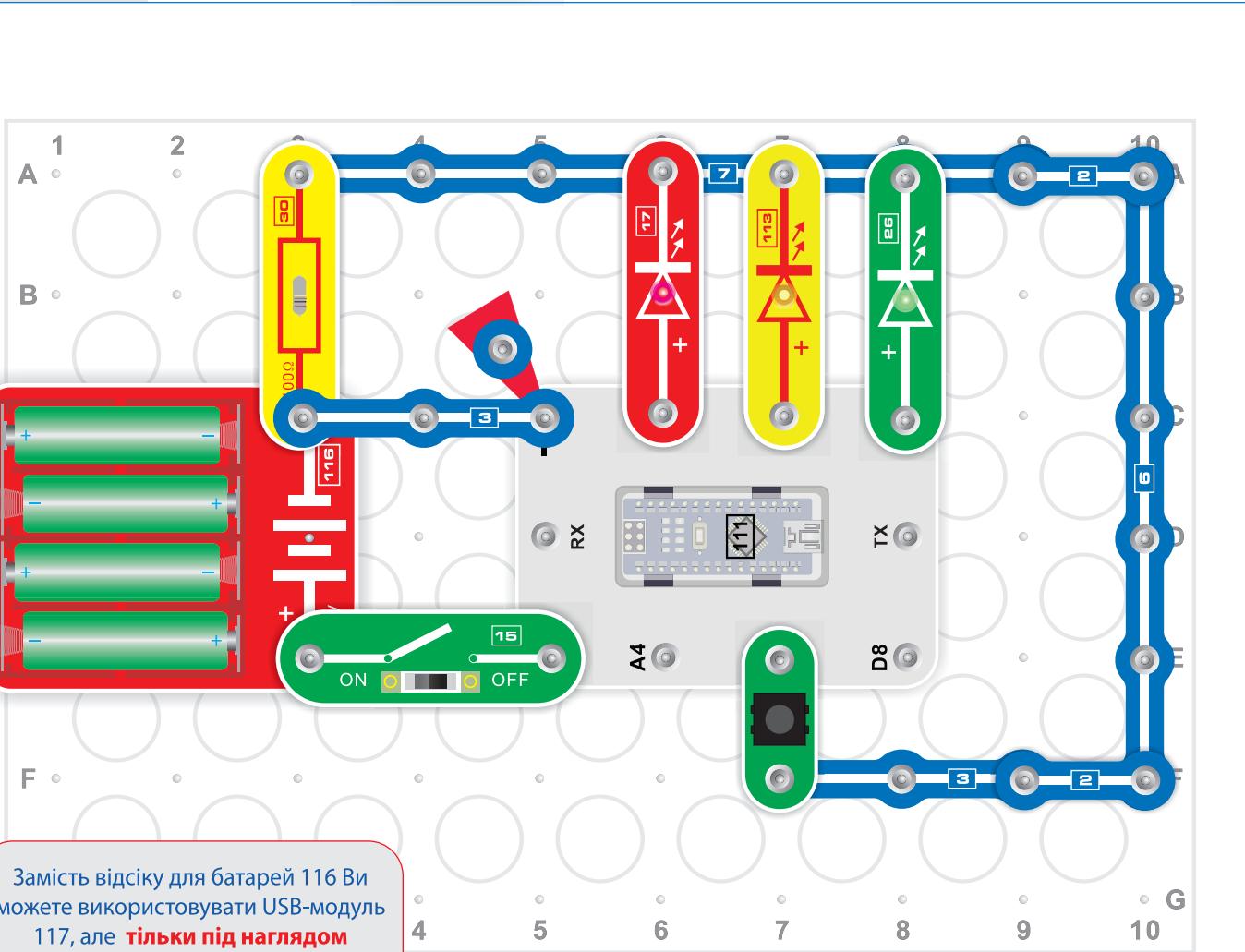
«Світлофор з вибором»

Ці три програми демонструють різні режими роботи кнопки.

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму.

Після завантаження програми від'єднайте кабель та увімкніть живлення — вимикач **15** в положенні ON.

Повний текст програми можна подивитися на флешці, що входить до комплекту.



**ZNATOK™**

## ПРОГРАМИ

### 01\_Control\_LED-17

У цій програмі відбувається послідовна зміна режимів роботи — кожне натискання вмикає наступний режим: **«Вогні, що біжать»** — **«Плавне увімкнення»** — **«Світлофор»**. Але такий вибір режиму не завжди буває зручним, коли їх багато.

### 01\_Control\_LED-19

Ця програма дозволяє послідовним натисканням на кнопку обрати один з режимів роботи світлофора: **«Стандартний»**, **«Ранковий»** (зі збільшеним часом роботи зеленого сигналу), **«Нічний»** (миготливий жовтий).

```
if (timing < 500)
{
    digitalWrite(LED_YELLOW, HIGH);
}
else if (timing < 1000)
{
    digitalWrite(LED_YELLOW, LOW);
}
else
{
    start_cycle = millis();
}
...

```

**Швидкість мережотіння жовтого**  
Змінити вказані параметри на 250 і 500 відповідно і жовтий в аварійному режимі буде світити в два рази частіше.

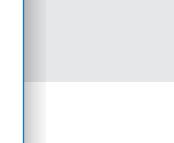
### 01\_Control\_LED-18

Ця програма дозволяє одразу обрати потрібний режим за допомогою кнопки:

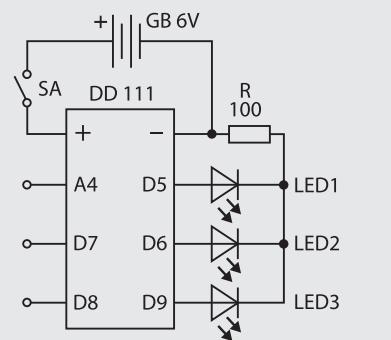
- 1 коротке натискання — **«Вогні, що біжать»**
- 2 коротких натискання — **«Плавне увімкнення»**
- 1 тривале натискання (більше 2 секунд) — **«Світлофор»**

Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля **111(Arduino)**!

## ЗМІШУЄМО КОЛЬОРИ



**ZNATOK™**

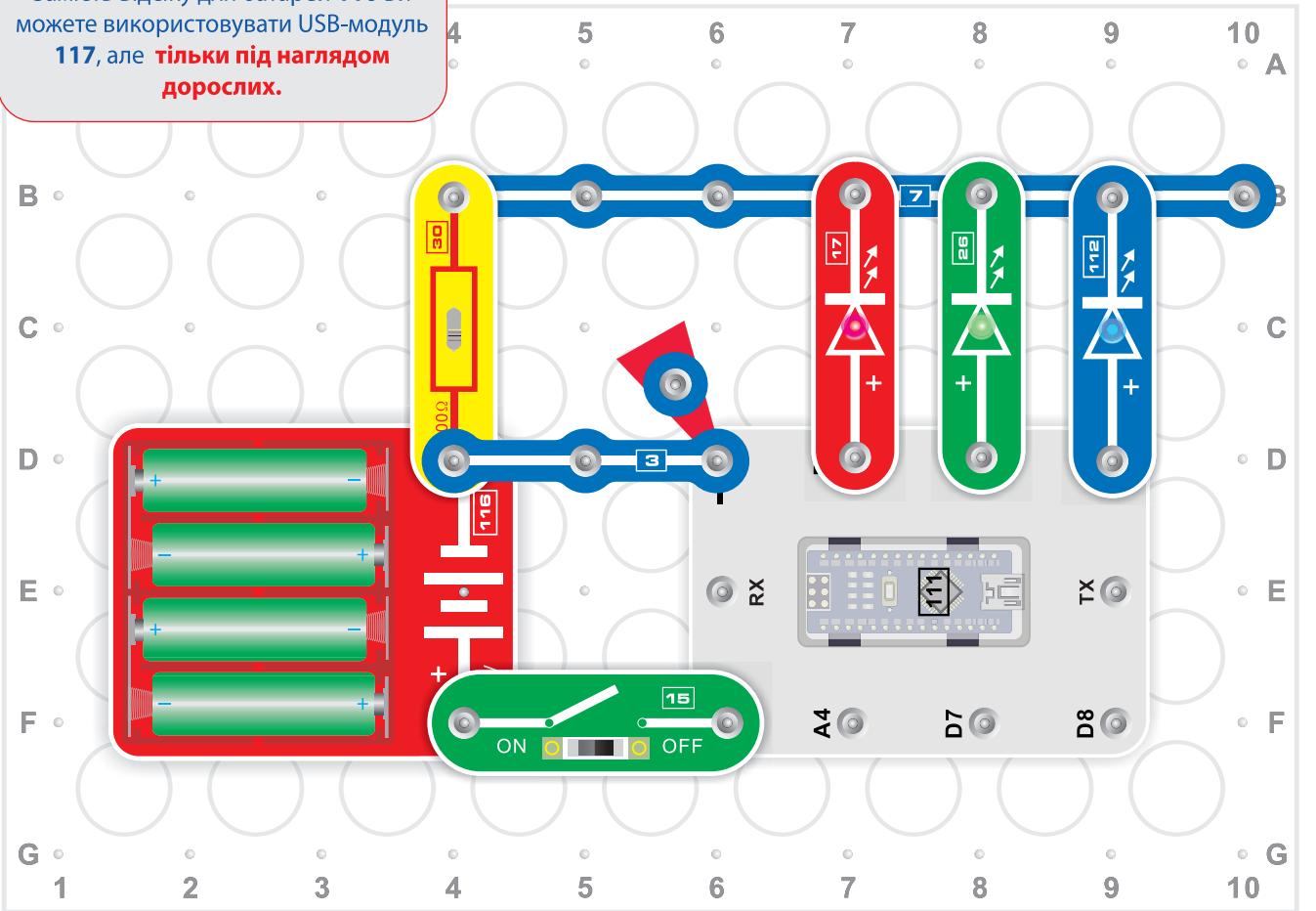


Важко уявити, що всі кольори на Вашому смартфоні, моніторі чи телевізорі утворюються лише з трьох кольорів — **червоний (Red)**, **зеленого (Green)** та **синього (Blue)**. Але це дійсно так!

Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 1 програма: **03\_RGB\_Mix-1** — «Три окремих кольори».

Підключіть модуль **111** до комп’ютера за допомогою USB-кабелю та завантажте одну з наведених програм. Після завантаження програми від’єднайте кабель та увімкніть живлення — вимикач **15** у положенні ON.

Замість відсіку для батарей **116** Ви можете використовувати USB-модуль **117**, але **тільки під наглядом дорослих.**



**ZNATOK™**

## ПРОГРАМИ

### 03\_RGB\_Mix-1

Ця програма призначена для опрацювання керування трьома основними кольорами — **червоний**, **зеленим** і **синім**. Після завантаження програми всі три світлодіоди почнуть вмикатися, імітуючи режим «Веселка». Нижче вказано, як можна відрегулювати затримку перемикання між кольорами та додати особливий колір у послідовність.

Кожен світлодіод має свою характеристику світіння, і якщо задавати їм одинаковий сигнал керування, то вони будуть світити по-різному. В цій програмі можна змінювати яскравість світіння кожного світлодіода, що дозволить в подальшому створювати потрібні Вам кольори.

**Затримка перемикання**  
Змініть це значення з **1000** до **500**, і кольори почнуть перемикатися швидше

...  
int switchDelay = 1000;  
...

Для цього необхідно додати наступні команди:

**Яскравість зеленого світлодіода**  
Змініть це значення від **0** (зелений світлодіод не світиться) до **255** (зелений світлодіод світиться з максимальною яскравістю)

...  
setRGB(255, 255, 255);  
delay(switchDelay);  
...

**Яскравість червоного світлодіода**  
Змініть це значення від **0** (червоний світлодіод не світиться) до **255** (червоний світлодіод світиться максимальною яскравістю)

...  
setRGB(255, 0, 0);  
delay(switchDelay);  
...

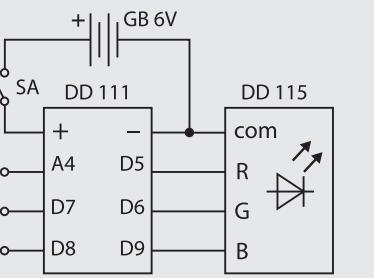
**Яскравість синього світлодіода**  
Змініть це значення від **0** (синій світлодіод не світиться) до **255** (синій світлодіод світиться з максимальною яскравістю)

...  
setRGB(0, 255, 255);  
delay(switchDelay);  
...

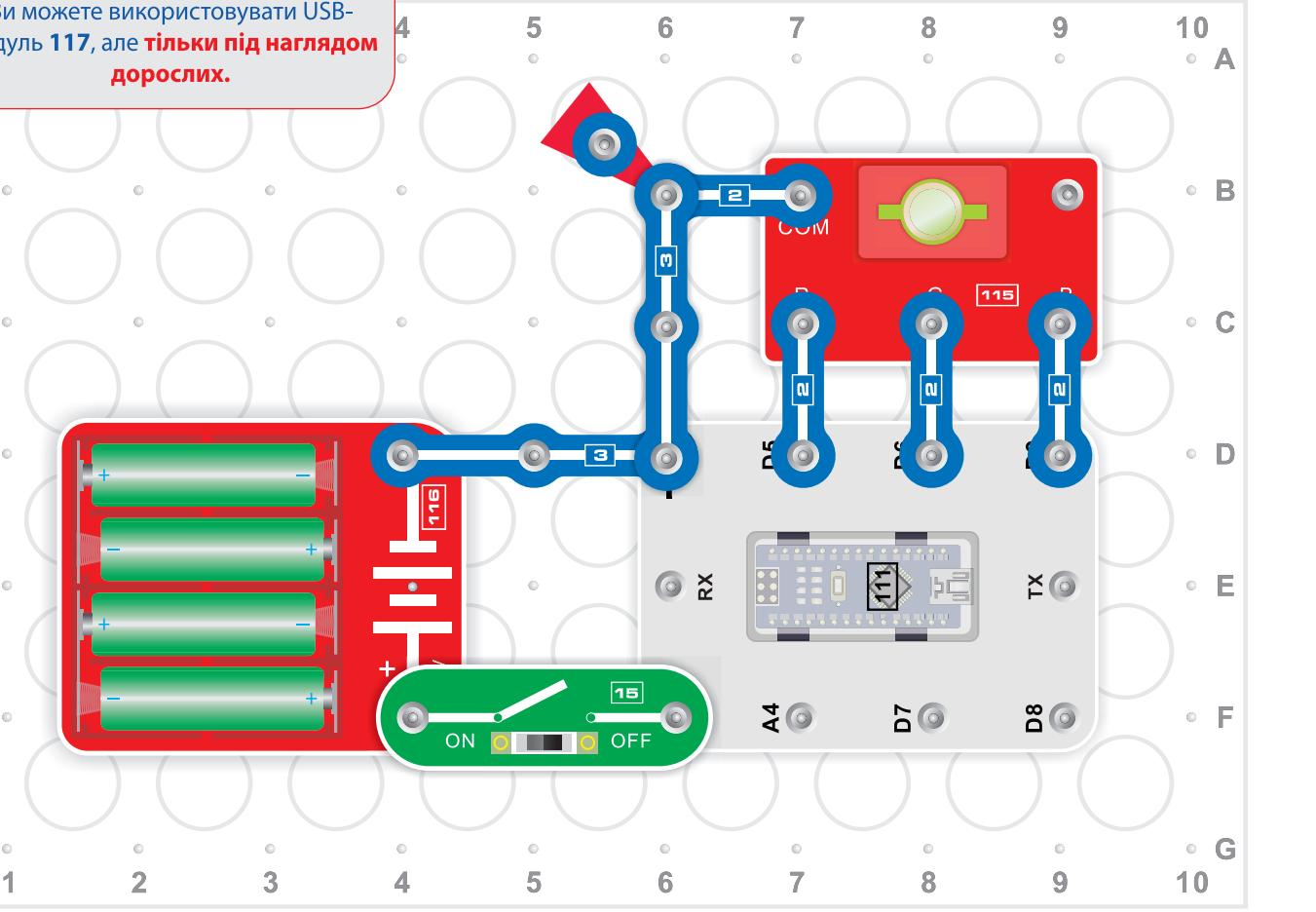
Вищенаведені команди треба розмістити ось в цьому місці програми:

У цій схемі не видно, як змішуються кольори — світлодіоди розміщені занадто далеко один від одного, а ось в наступній схемі, де всі три світлодіоди об’єднані в одному корпусі, можна буде це побачити.

Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля **111(Arduino)**!



Замість відсіку для батарей **116** ви можете використовувати USB-адаптер **117**, але **тільки під наглядом дорослих.**



Зберіть коло, схему якого зображенено на рисунку. За допомогою тримача встановіть на модуль **115** прозорий циліндр. До цієї схеми додаються **2** програмами:

## **03\_RGB\_Mix-2** — «Веселка».

**03\_RGB\_Mix-3** — «Воля випадку». Всі кольори будуть відтворюватися у випадковому режимі.

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму.

Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач **15** в положенні ON.

48

## ПРОГРАМ

## RGB\_Mix-2

завантаження програми **RGB**-світлоідод буде працювати в «Веселка» - послідовне увімкнення кольорів від червоного до зеленого. Зверніть увагу, що дана програма відрізняється від програми **01\_RGB\_Mix-1** лише іншими величинами керуючих змінних. Програми для формування кольорів – це пов’язано з розподілом світлодіодів всередині корпусу **RGB** та світлодіоди у власних колодах, навіть маючи один колір, мають різні характеристики. На зображені, як можна відрегулювати затримку перемикання

цювати в між кольорами та додати власний колір у пос.

### Затримка перемикач

Змініть це значення на **500**, і кольори почнуть перемікатися швидше

...  
int switchDelay =

програмі можливо створювати власні кольори. Для цього потрібно додати наступні команди:

**G(reen) складова**

Змініть це значення з **0**  
світлодіод не світиться  
(зелений світлодіод світиться  
максимальною яскравістю)

**G(reen) складова кольору**  
Змініть це значення з **0** (зелений світлодіод не світиться) до **255** (зелений світлодіод світиться з максимальною яскравістю)

**B(lue) складова кольору**  
Змініть це значення з **0** (синий світлодіод не світиться) до **255** (синий світлодіод світиться з максимальною яскравістю)

наведені команди треба розмістити ось в цьо-  
програми:

```
... setRGB(0, 0, 255);  
delay(switchDelay);
```

setRGB(255, 0, 255);

кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте оновлену програму до модуля **111** (Arduino)!

## ПРОГРАМИ

## 03\_RGB\_Mix-3

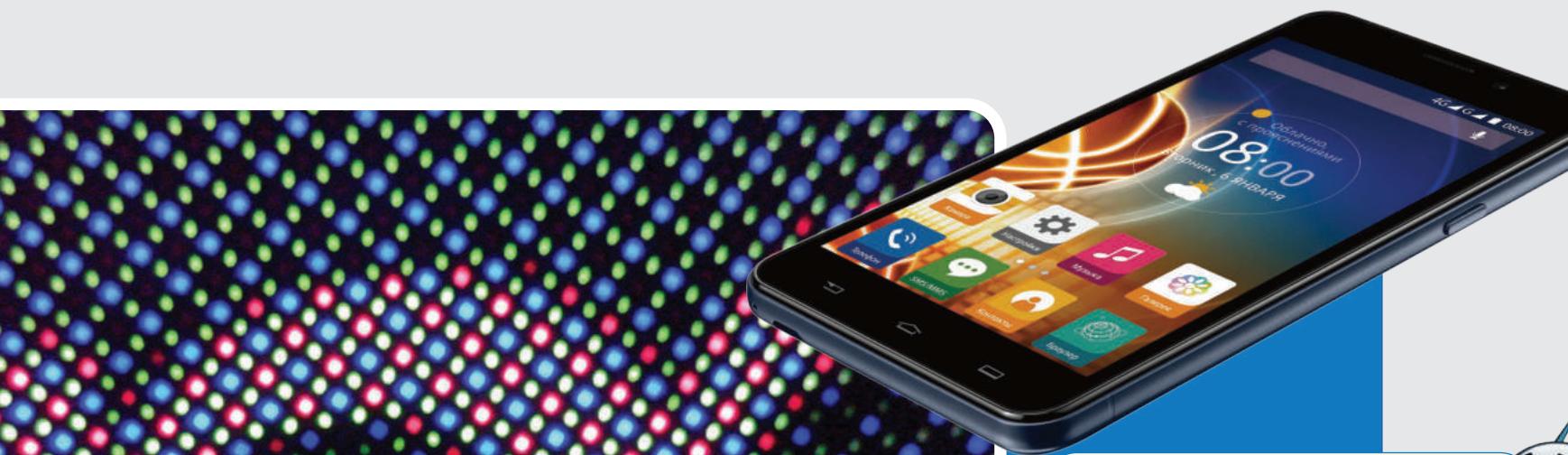
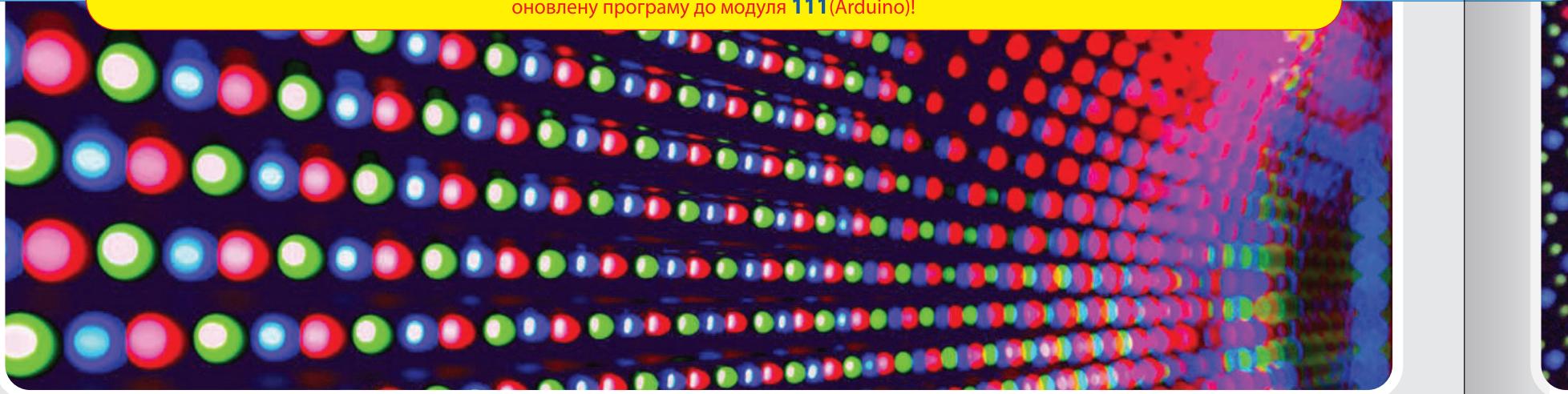
Після завантаження програми RGB-світлодіод буде випадково змінювати свій колір. Нижче зображено, як можна відрегулювати швидкість перемикання.

## Затримка перемикання

Змініть це значення з **250** до **5000** – колір почне змінюватися кожні 5 секунд. Спробуйте вгадати, як отримати той чи інший колір!

```
...
void loop()
{
    int switchDelay = 250;
    setRGB(random(256), random(256), random(256));
    delay(switchDelay);
}
...
```

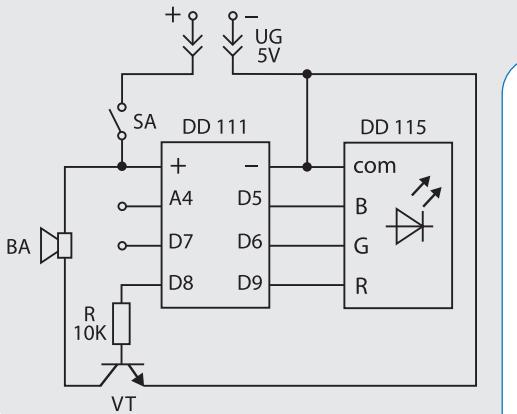
Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля 111(Arduino)!



## ЦЕ ЦІКАВО

Екран смартфону під мікроскопом. Можна побачити, що зображення складається з крапок – світлодіодів, що світяться.





Замість USB-модуля 117 Ви можете використовувати відсік для батарей 116, але **тільки під наглядом дорослих.**

Зберіть коло, схему якого зображено на рисунку. До цієї схеми додаються 2 програми:

#### 04\_Color\_Music-1

«Музична веселка»

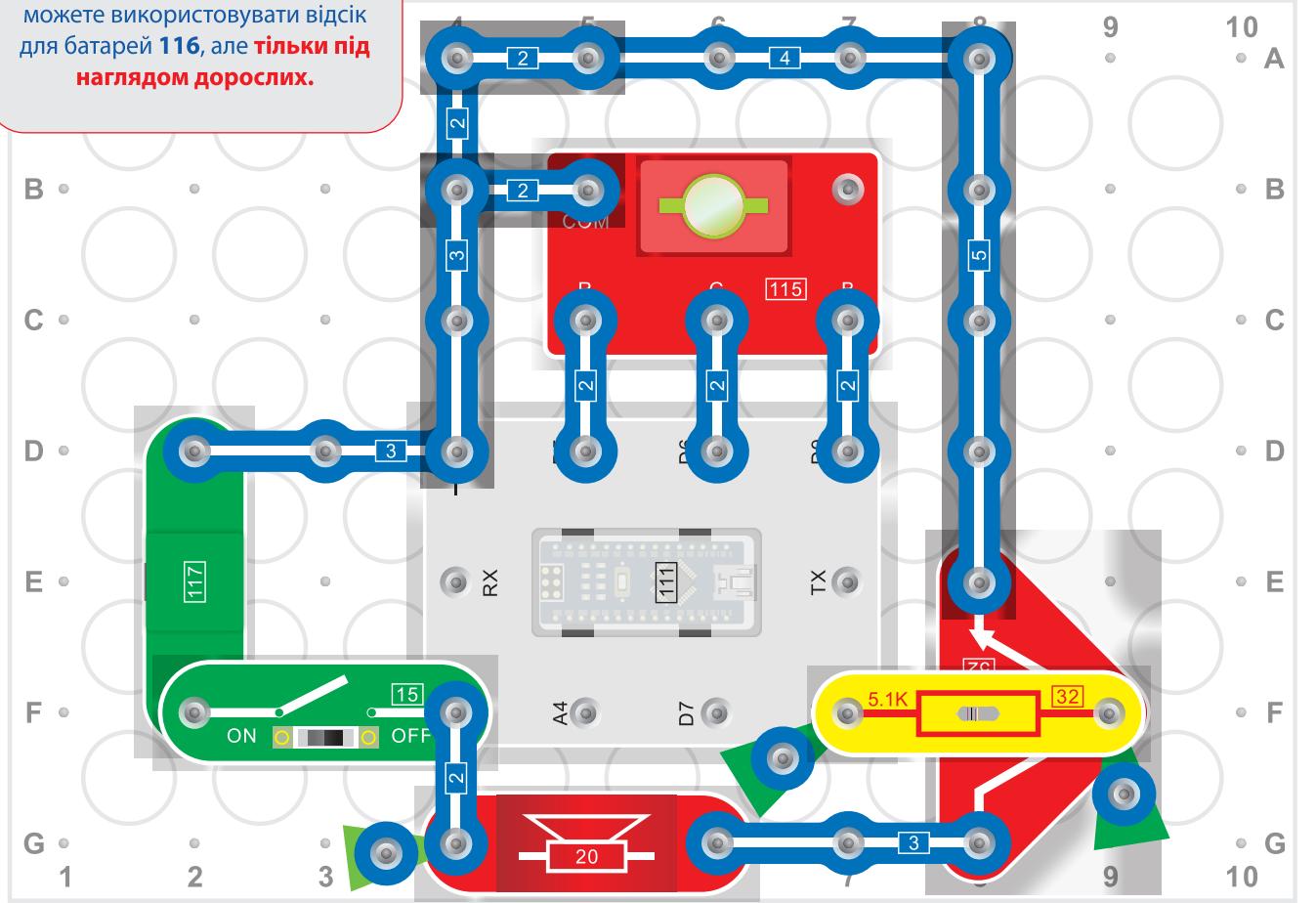
#### 04\_Color\_Music-2

«Світлова какофонія»

Підключіть модуль 111 до комп'ютера за допомогою USB-кабелю та завантажте програму.

Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач 15 в положенні ON.

Замість USB-модуля 117 Ви можете використовувати відсік для батарей 116, але **тільки під наглядом дорослих.**



## ПРОГРАМИ

### 04\_Color\_Music-1

Після завантаження програми світлоідодний модуль 115 почне послідовно світити сімма кольорами веселки, а динамік відтворювати сім нот. Потім цикл повторюється.

Нижче наведений фрагмент програми та зображене, як можна змінити швидкість перемикання, а також змінити октаву, з якої починають відтворюватися ноти. Цілком програму можна подивитись на флешці, що входить до комплекту.

#### Затримка перемикання

Змініть дане значення з **300** на **500** і переконайтесь, що кольори з нотами стали перемікатися швидше.

```
int switchDelay = 300;
int minFrequency = 400;
int maxFrequency = 1000;
```

#### Номер октафи

Змініть **0** на **1** і переконайтесь в тому, що ноти стали відтворюватися на октаву вище. Зверніть увагу, що в даній програмі передбачається використання лише двох октав, таким чином, зміна даного параметра на інші значення не дозволить програмі працювати правильно.

#### Мінімальна частота

Змініть даний параметр **3 400** на **150** і переконайтесь в тому, що серед усіх нот стали тряплятися дуже низькі

#### Максимальна частота

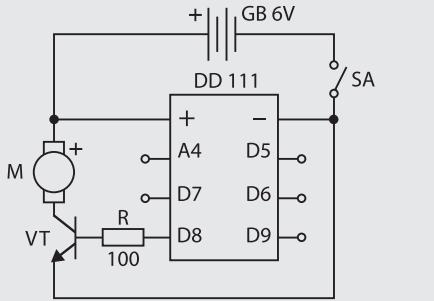
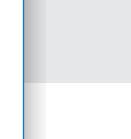
Змініть дане значення **3 1000** на **5000** і переконайтесь, що серед усіх нот стали тряплятися дуже високі.

Після кожної зміни в програмі зберігайте оновлену версію в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля 111(Arduino)!



## ЦЕ ЦІКАВО

Багато відомих композиторів, наприклад, Скрябін, Римський-Корсаков, Чурльоніс, Мессіан, намагались пов'язати музичну ноту з певним кольором. У всіх виходило по-різному. В цьому керівництві були взяті сім нот (**До-Ре-Мі-Фа-Соль-Ля-Сі**) та прив'язані до семи кольорів веселки (**Червоний-Помаранчевий-Жовтий-Зелений-Блакитний-Синій-Фіолетовий**). Ви можете створювати свої комбінації.



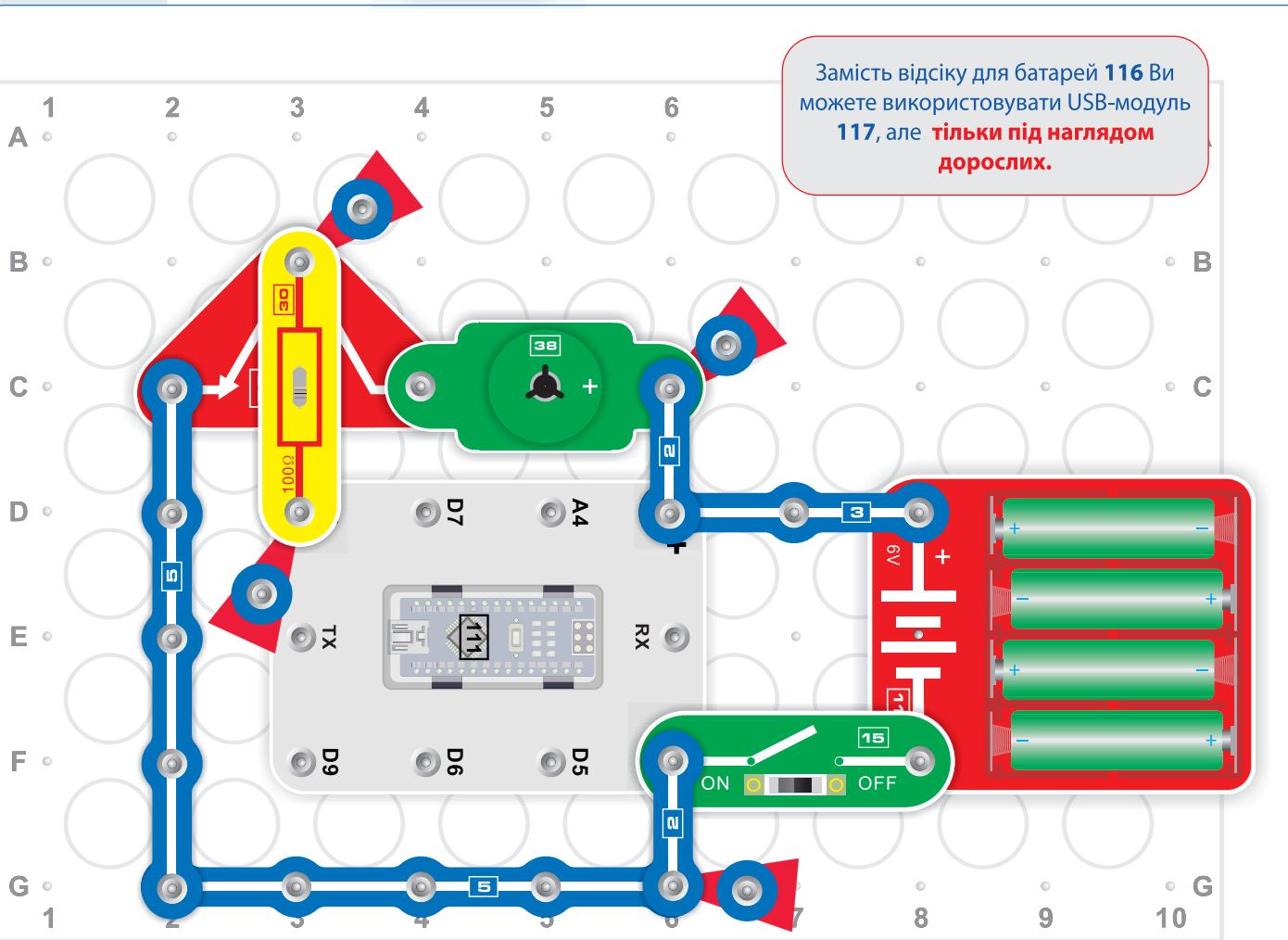
Зберіть коло, схему якого зображенено на рисунку, але не встановлюйте пропелер на електродвигун. До цієї схеми додається **1** програма:

05\_Motor\_Control-1 —

## «Транзистор на допомогу»

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач **15** в положенні ON.

Не забувайте, що максимальний струм, який може виробляти на виходах модуль **111**, не перевищує **40 mA**, а для роботи електродвигуна **38** потрібен струм у кілька разів більший. Тому в якості підсилювача струму використовується транзистор **52**.



54

ПРОГРАММА

5 Motor Control-1

Після завантаження програми електродвигун 5 від вмикається на 2 секунди та вимикається на 2 секунди. Для даного типу транзистора, щоб вимкнути двигун, на вивід D8 подається низький рівень напруги (LOW), а для увімкнення – високий (HIGH). Так як транзистор D8 не має вбудовану функцію ШІМ, то можливе лише увімкнення-вимкнення електродвигуна, без плавного змінення швидкості обертання. Зверніть увагу, що на відміну від більшості попередніх програм, ця програма виконується один раз, а не циклічно, і для повторного запуску потрібно вимкнути і увімкнути живлення (OFF-ON).

Нижче наведений фрагмент програми та зображені, можна змінити кількість увімкнень та вимкнень кротодвигуна, а також тривалість часу роботи при окремому увімкненні.

Цілком програму можна подивитись на флешці, що входить до комплекту.

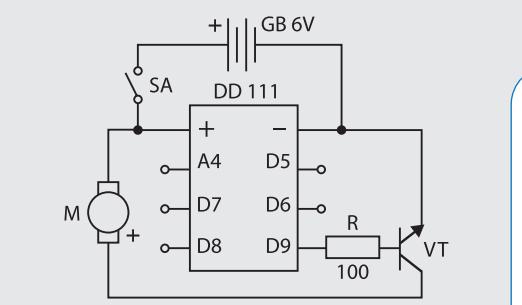
## ДЛЯ ПРОСУНУТИХ

з розділу «**КЕРУВАННЯ СВІТЛЮДІОДАМИ**» було описано, як програмними засобами організувати плавне регульовання швидкості обертання електродвигуна.

Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте оновлену програму до модуля **111**(Arduino)!

```
void setup() {
    ...
    pinMode(MOTOR_PIN, OUTPUT);
    //У самому початку вимикаємо моніторинг
    digitalWrite(MOTOR_PIN, LOW);

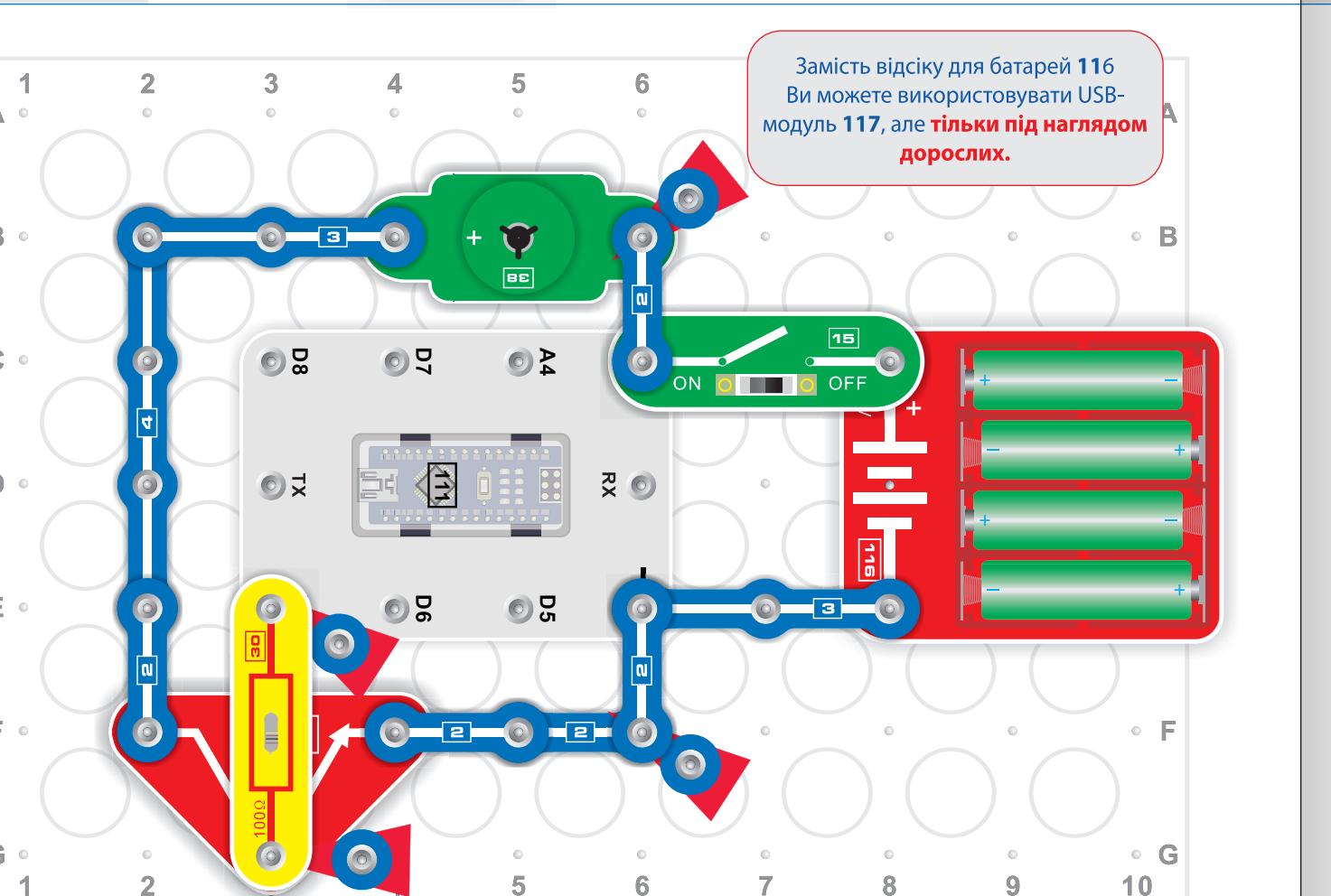
    int switchDelay = 2000;
    for (int i = 0; i < 5; i++) {
        {
            digitalWrite(MOTOR_PIN, LOW);
            delay(switchDelay);
            digitalWrite(MOTOR_PIN, HIGH);
            delay(switchDelay);
        }
    }
}
```



На відміну від попередньої схеми, тут можна регулювати швидкість обертання електродвигуна завдяки вбудованій функції **ШІМ** на виводі D9. Зберіть коло, схему якого зображенено на рисунку, але не встановлюйте пропелер на електродвигун. До цієї схеми додається **1** програма:

## **05\_Motor\_Control-2 —** «ШІМ на допомогу транз

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач **15** в положенні **ON**.



у для батарей 116  
ористовувати USB-  
**ільки під наглядом**  
**ослих**

05\_Motor\_Control-2

Після завантаження програми електродвигун вмикається на 1 секунду та вимикається на 3 секунди, знову вмикається та вимикається, потім вмикається на 1 секунду та плавно, завдяки використанню **ШІМ**, зменшує свою швидкість аж до повної зупинки. Щоб закрити NPN-транзистор і вимкнути електродвигун, на вивід D9 подається низький рівень напруги (**LOW**), а для відкриття транзистора та увімкнення електродвигуна – високий (**HIGH**). Зверніть увагу, що на відміну від більшості попередніх програм, ця програма виконується один раз, а не циклічно, і для повторного запуску потрібно вимкнути і увімкнути живлення (**OFF-ON**).

Нижче наведений фрагмент програми та зображено, як можна змінити кількість увімкнень та плавність зупинки електродвигуна, а також тривалість часу роботи при кожному увімкненні.

Цілком програму можна подивитись на флешці, що входить до комплекту.

КЕРУВАННЯ ЕЛЕКТРОДВИГУНОМ (ЗАКІНЧЕННЯ)

## ПРОГРАМИ

#### Кількість вимкнень

**Іншість уміння**  
Ільшіть даний параметр  
**5** і переконайтесь, що  
електродвигун став вмикатись  
а не 3 рази

```
i < 3; i++)
```

IWrite(MOTOR\_PIN,  
(3000);

Write(  
    (1000);

EE-30

gWrite(

2018

qWrite()

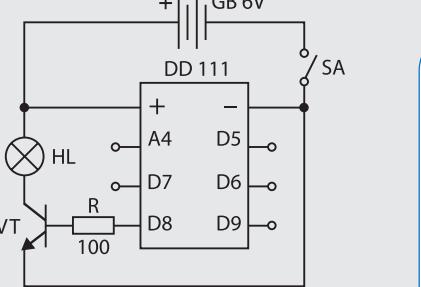
(800),

Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте за-  
новлену програму до модуля **111**(Arduino)!

## КЕРУВАННЯ ЛАМПОЮ РОЗЖАРЮВАННЯ



**ZNATOK™**

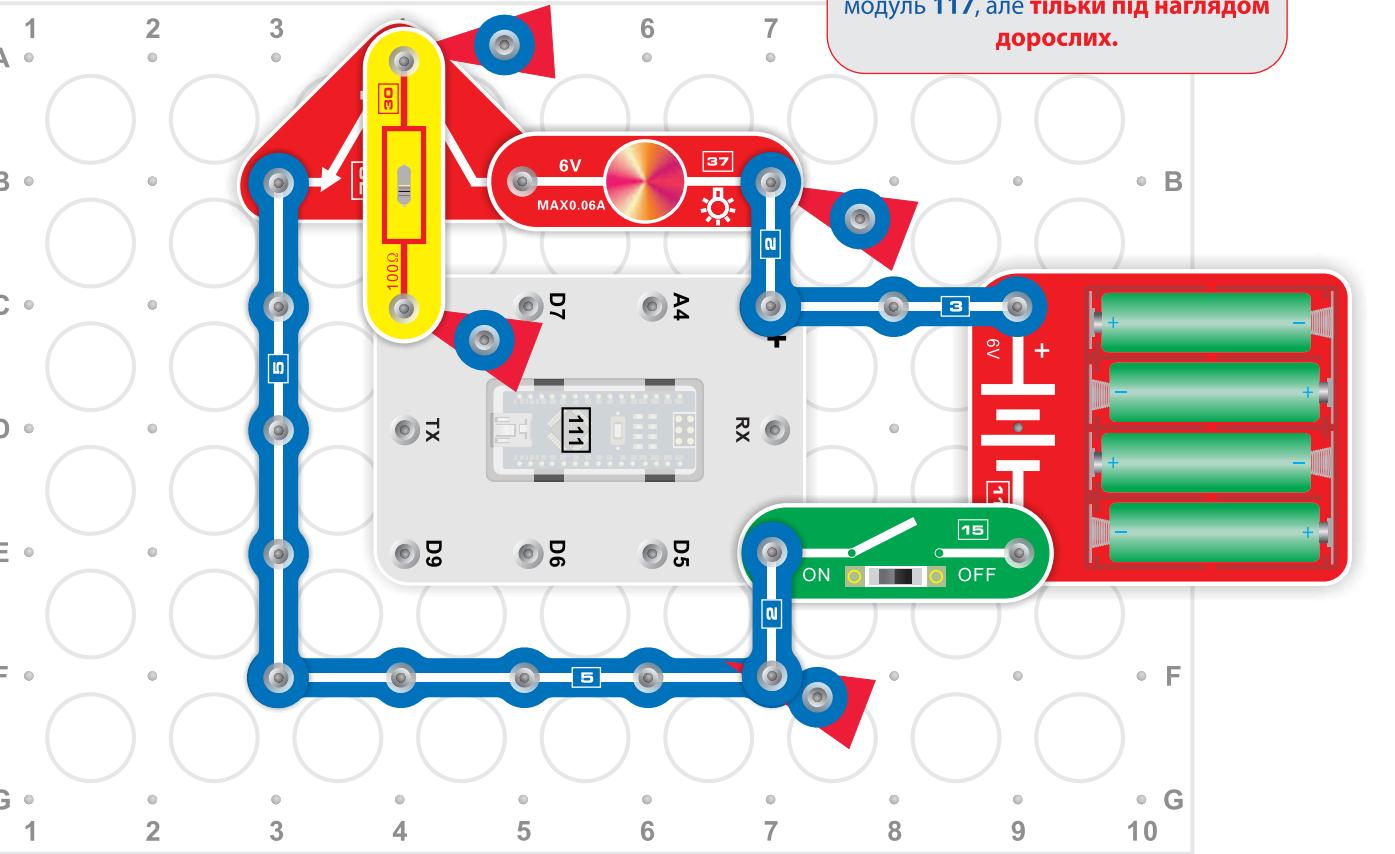


Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 1 програма

### 06\_Lamp\_Control-1 «Керування NPN-транзистором»

Підключіть модуль 111 до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач 15 в положенні ON.

Не забувайте, що максимальний струм, який може виробляти на виходах модуль 111, не перевищує 40 мА, а для роботи лампи 38 потрібен струм 60 мА. Тому, в якості підсилювача струму, використовується транзистор.



**ZNATOK™**

## ПРОГРАМИ

### 06\_Lamp\_Control-1

#### 06\_Lamp\_Control-1

У цій програмі лампа, підключена через NPN-транзистор до виводу D8 вмикається та вимикається на 2 секунди. Нижче наведений фрагмент програми, у якому зображено, як можна не тільки змінити частоту вимикання лампи, але й зменшити яскравість лампи, не використовуючи функцію вбудованого ШІМ.

Цілком програму можна подивитись на флешці флешці, що входить до комплекту.

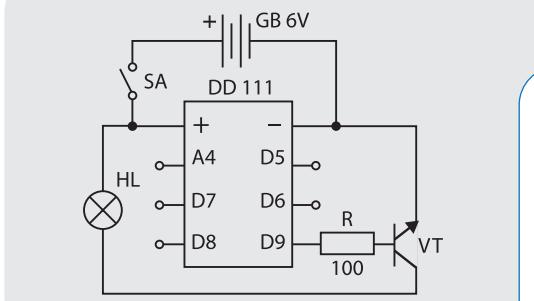
```
...
#define LAMP_PIN 8
void setup()
{
    pinMode(LAMP_PIN, OUTPUT);
}
void loop()
{
    int switch_delay = 2000;
    digitalWrite(LAMP_PIN, HIGH);
    delay(switch_delay);
    digitalWrite(LAMP_PIN, LOW);
    delay(switch_delay);
}
...
```

#### Час перемикання

Замініть даний параметр на 1000 і переконайтесь, що тепер лампа вмикається і вимикається на одну секунду. Ви можете зменшити це значення до 5 і переконатися, що лампа стала світити тьмяніше

Після кожної зміни в програмі зберігайте оновлену версію в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля 111(Arduino)!

## КЕРУВАННЯ ЛАМПОЮ РОЗЖАРЮВАННЯ (Продовження)

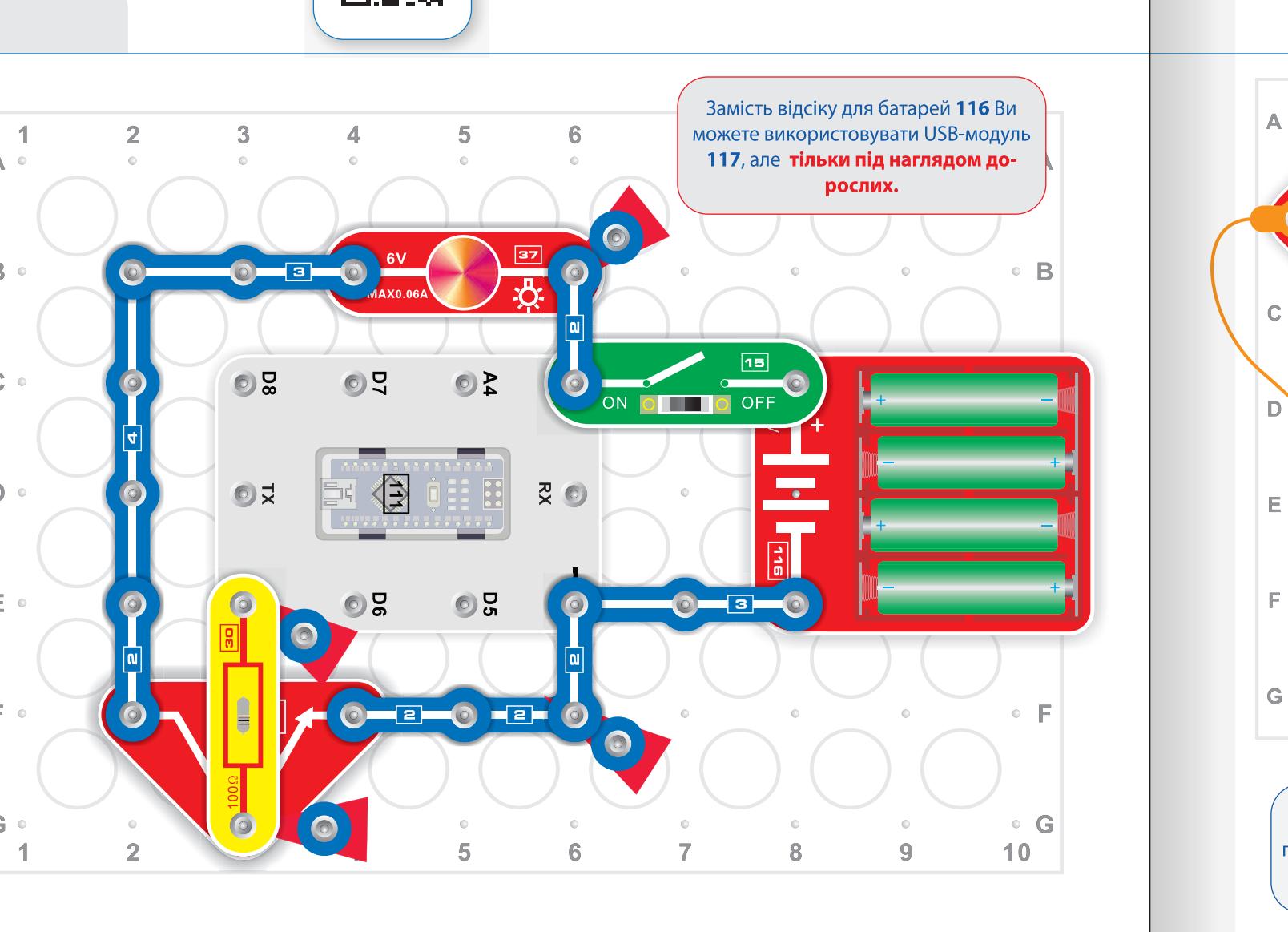


Попередня схема дозволяла тільки вмикати та вимикати лампу з однаковою яскравістю на заданий проміжок часу. Данна схема додатково дозволяє змінювати яскравість світіння лампи. Цього можна досягнути завдяки використанню виводу D9 із будованою функцією ШІМ.

Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 1 програма

**06\_Lamp\_Control-4** — «Керування ШІМ NPN-транзистором»

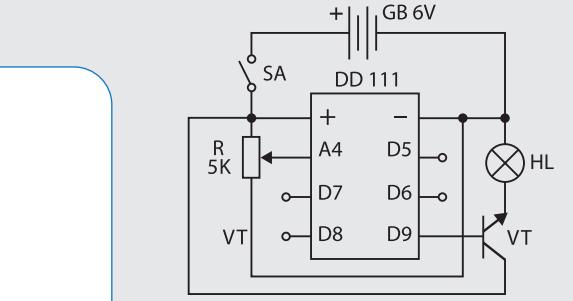
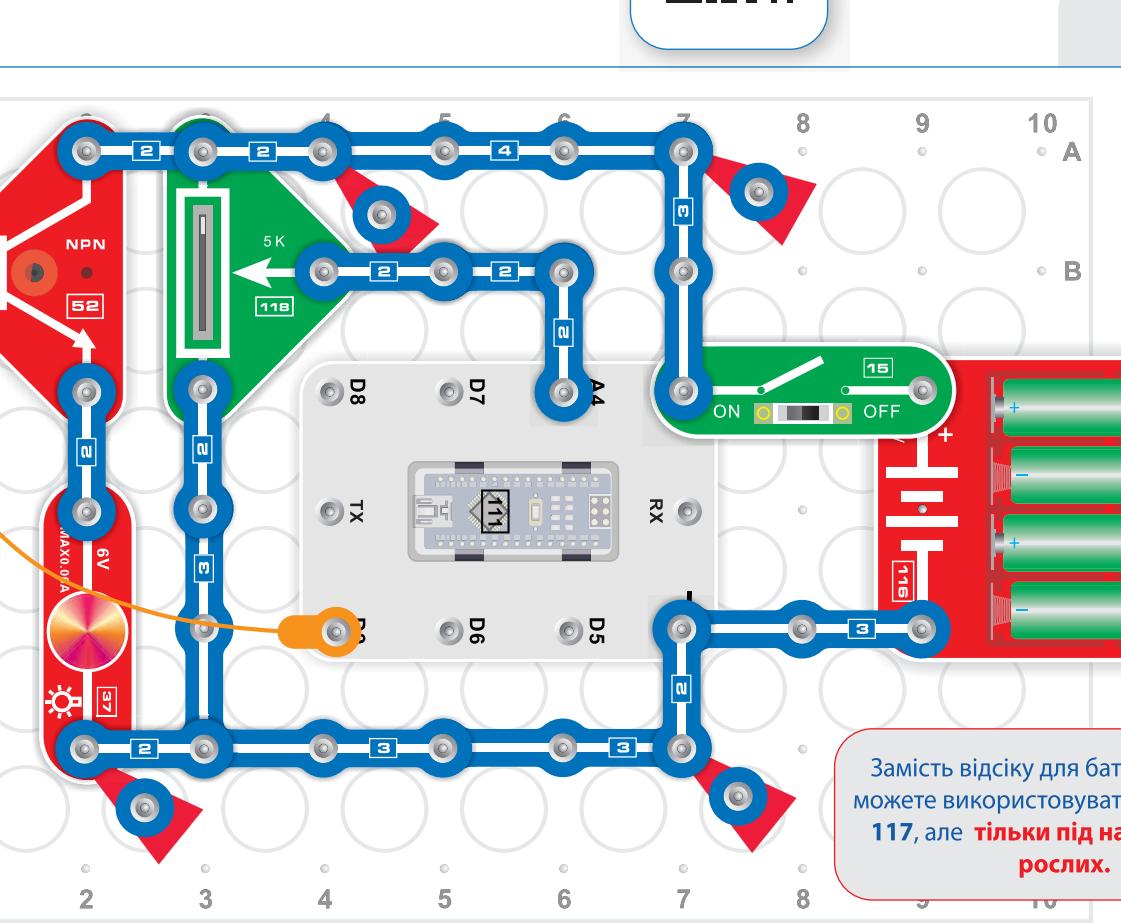
Підключіть модуль 111 до комп’ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від’єднайте кабель та увімкніть живлення — вимикач 15 в положенні ON. Лампа почне плавно збільшувати свою яскравість протягом 3 секунд, а потім різко згасати.



Програми можна подивитись на флешці, що входить до комплекту.

Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 3 програмами:

**06\_Lamp\_Control-5** — «Лінійна залежність». Під час пересування движка догори, яскравість лампи буде збільшуватись. Але це можна було б зробити й без мікроконтролера! А ось наступні програми відображають, як можна змінювати режими роботи однієї і тієї ж схеми.



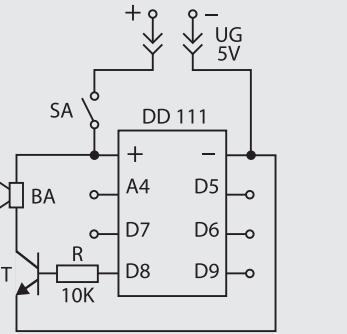
**06\_Lamp\_Control-6** — «Нелінійна залежність». Під час пересування движка догори, яскравість лампи буде спочатку збільшуватись, а потім зменшуватись. Жоден зі змінних резисторів не має такої характеристики!

**06\_Lamp\_Control-7** —

«Перевантаження». Під час пересування движка догори, яскравість лампи буде збільшуватись, але якщо заданий рівень буде перевищений, лампа переїде до режиму миготіння — попереджуючи про небезпеку!

Підключіть модуль 111 до комп’ютера за допомогою USB-кабелю та завантажте одну з програм. Після завантаження програми від’єднайте кабель та увімкніть живлення — вимикач 15 в положенні ON.

## СИНТЕЗATORI ЗВУКІВ



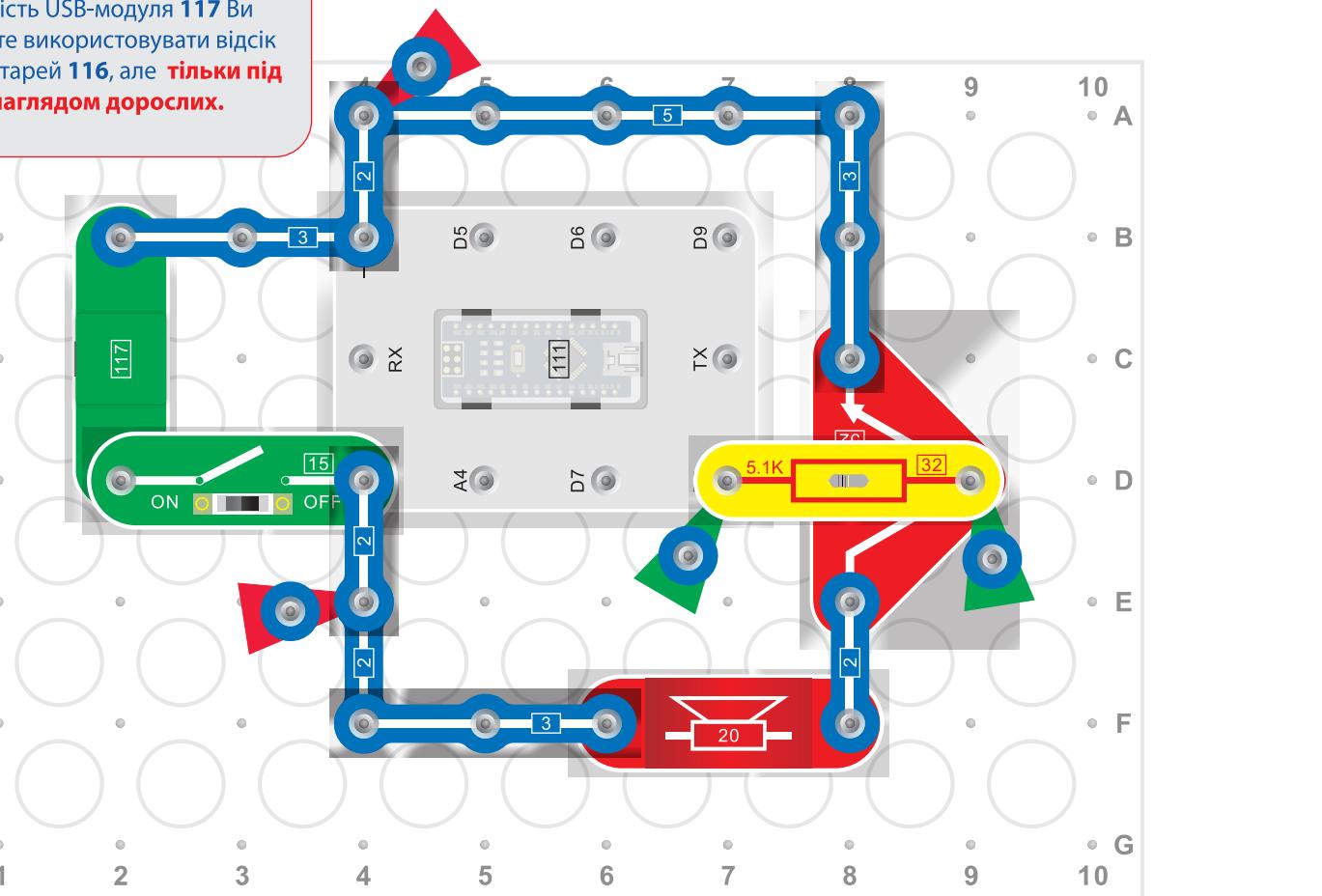
Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 1 програма:

**07\_Synth-1** —  
«За заданою програмою»

Підключіть модуль 111 до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач 15 в положенні ON. Динамік почне відтворювати звуки, синтезуючи їх відповідно до програми.

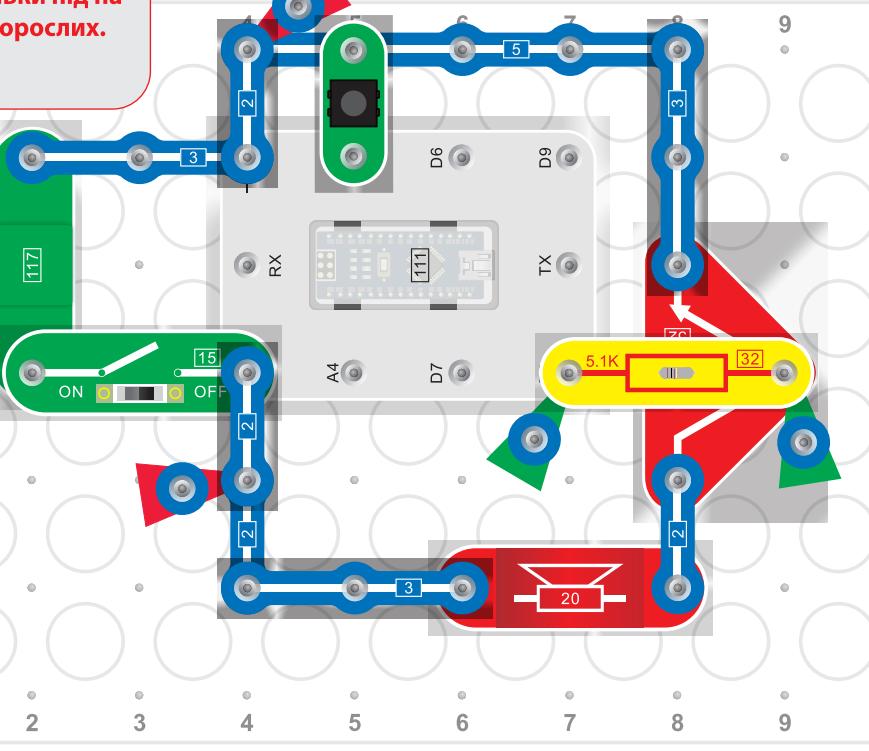
Транзистор 52 використовується як підсилювач.

Замість USB-модуля 117 Ви можете використовувати відсік для батарей 116, але **тільки під наглядом дорослих.**

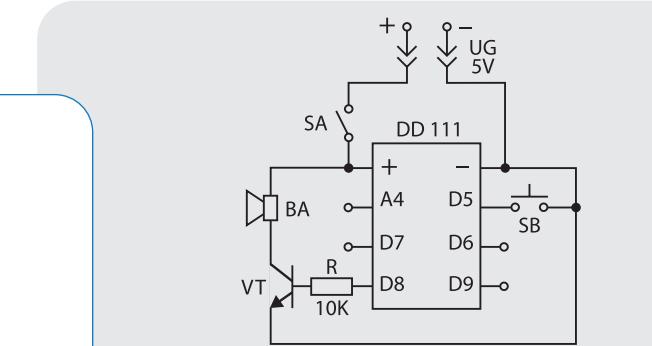


## СИНТЕЗATORI ЗВУКІВ

Замість USB-модуля 117  
Ви можете використовувати відсік  
для батарей 116, але **тільки під на-  
глядом дорослих.**



Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 3 програми:  
**07\_Synth-2a-S** — «Дверний дзвінок». Після короткого натискання на кнопку буде працювати, доки не вимкнуть живлення схеми.

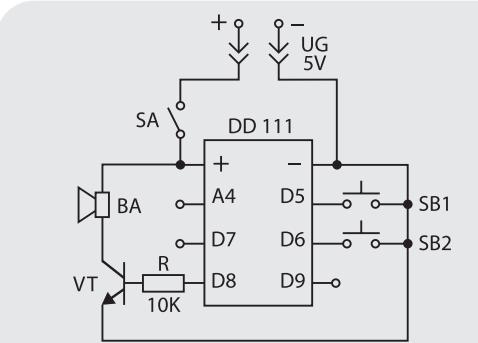


**07\_Synth-2c-S** — «Вхідний дзвінчик». Після короткого натискання на кнопку буде звучати «дзвінчик». Якщо замінити кнопку на геркон, то вийде схема, яка працює в магазинах, попереджуючи продавців, що до магазину зайшов покупець.

Підключіть модуль 111 до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач 15 в положенні ON. Натисніть на кнопку.

### ЦЕ ЦІКАВО

Зара, у більшості випадків, музичні інструменти налаштовують за нотою ЛЯ першої октави, яка, як було прийнято аж в 1836 році, дорівнює 440 Гц. Але так було не завжди. За сто років до цього нота ЛЯ дорівнювала 405 Гц, потім 419 Гц. Наполеон наказав ввести стандарт у 435 Гц, та у Франції досі, в деяких випадках, інструменти налаштовують по «наполеонівські».



Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається **2** програми:

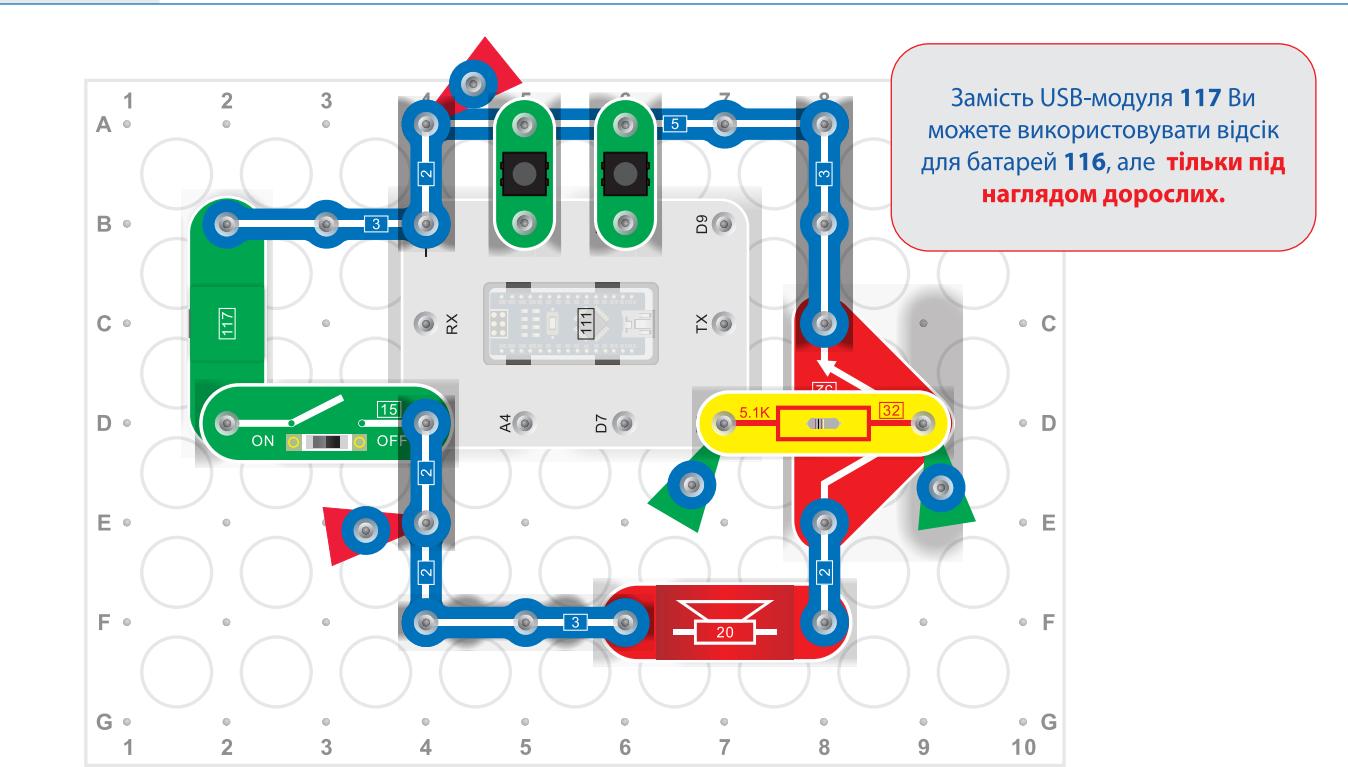
#### 07\_Synth-2d —

Управління звуками за допомогою двох кнопок. Варіант 1.

#### 07\_Synth-2e —

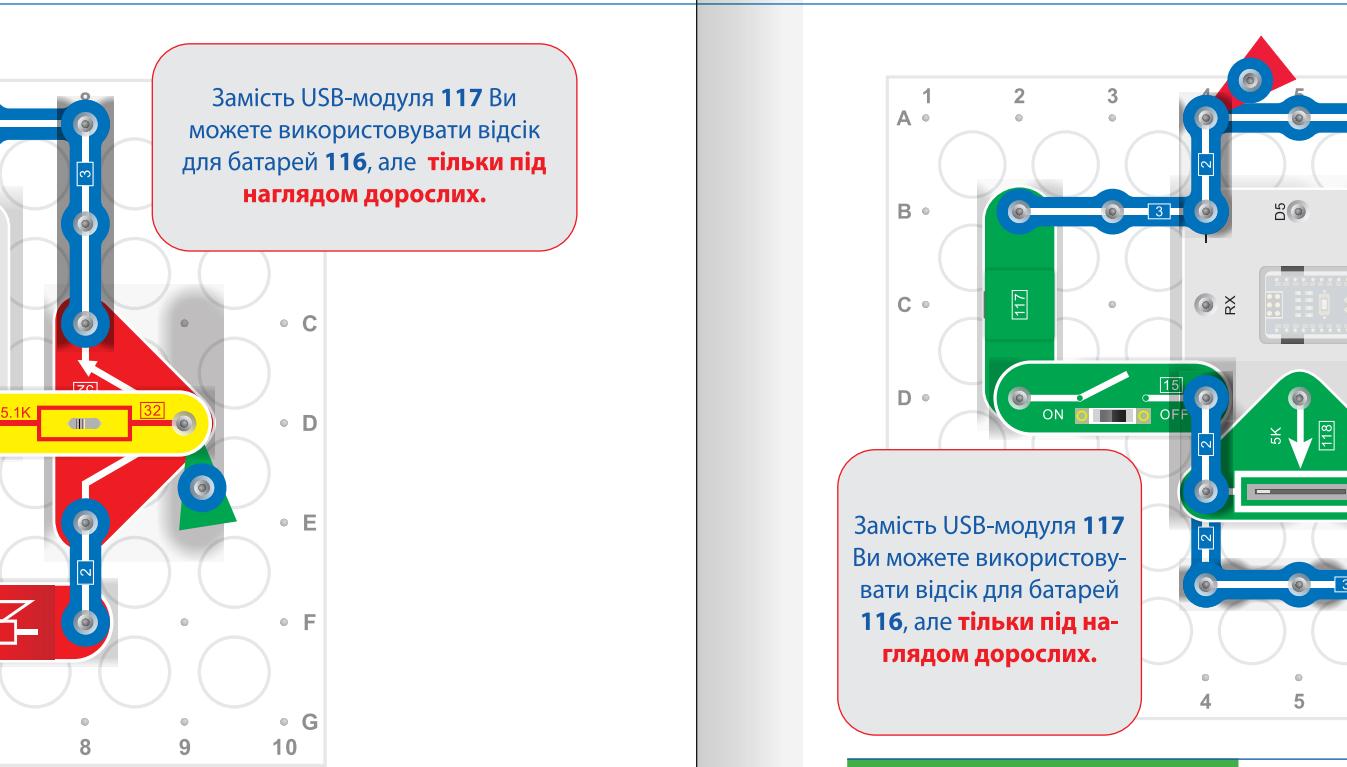
Управління звуками за допомогою двох кнопок. Варіант 2.

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму **07\_Synth-2d**. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач **15** в положенні ON.



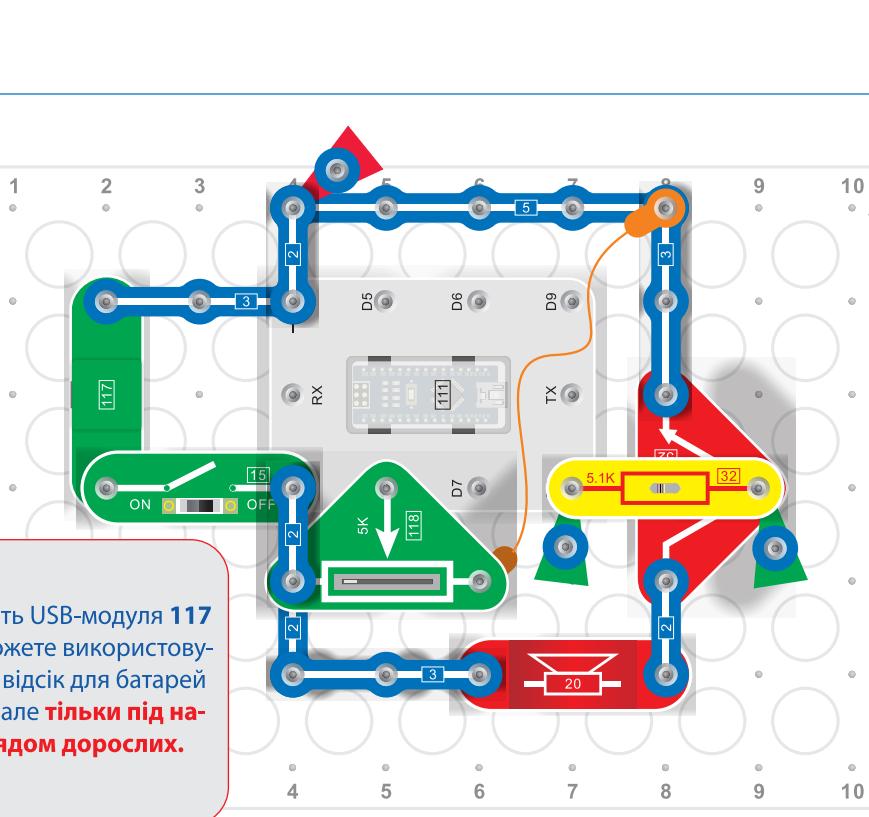
#### 07\_Synth-2d

Ця програма дозволяє змінювати висоту тону переривчатих звукових сигналів за допомогою двох кнопок. При натисканні на одну кнопку обирається одна із заздалегідь заданих частот відтворення звукових сигналів, а натискання на іншу кнопку – висоту тону.



#### 07\_Synth-2e

Ця програма задає інший спосіб керування за допомогою двох кнопок. При натисканні на одну кнопку обирається одна із заздалегідь заданих частот відтворення звукових сигналів, а натискання на іншу кнопку – висоту тону.



#### 07\_Synth-3

У цій програмі за допомогою резистора регулюється частота звуку, який відтворює динамік. Нижче наведений фрагмент програми, у якому зображене, як можна змінювати мінімальну та максимальну частоту звуку.

```
...
#define MIN_FREQUENCY 400
#define MAX_FREQUENCY 3000
...
```

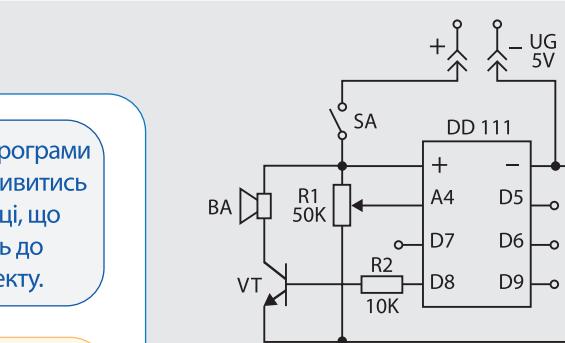
Повністю програми можна подивитись на флашці, що входить до комплекту.

#### Мінімальна частота

Замініть даний параметр на **100** і переконайтесь, що в одному з крайніх положень реостата звук став нижчим - **100 Гц** замість **400 Гц**.

#### Число ввімкнень

Замініть даний параметр до **6000** і переконайтесь, що при переміщенні реостата в крайнє положення звук став значно вищим - **6 кГц** замість **3 кГц**

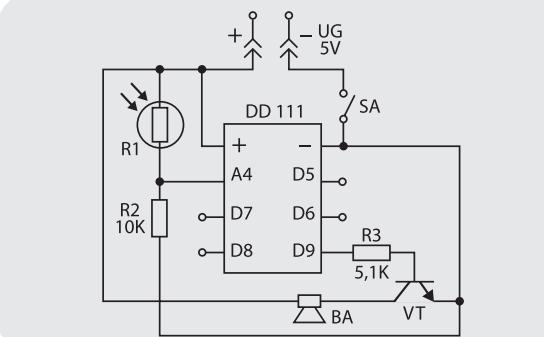


Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додаються **2** програми:

**07\_Synth-4** — «Три мелодії». Ця програма дозволяє за допомогою змінного резистора обрати одну з трьох відтворюваних мелодій – при крайньому лівому, центральному та крайньому правому положенні движка.

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач **15** в положенні ON. Змінюйте положення движка резистора – і спробуйте награти якусь мелодію.

## СИНТЕЗATORI ЗВУКІВ (ЗАКІНЧЕННЯ)



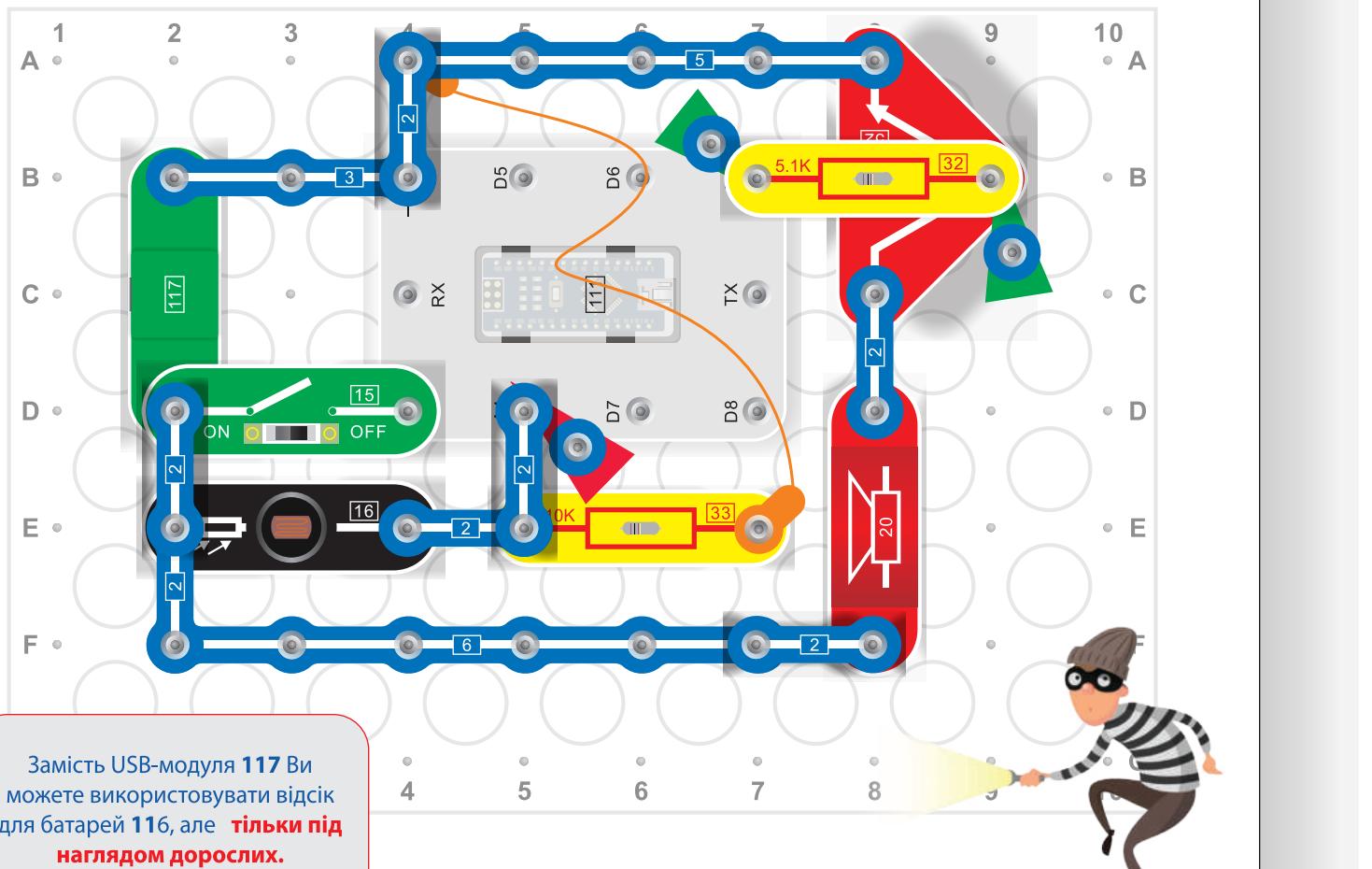
Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додаються **2** програми:

**07\_Synth-5** — «Звуки світла». Якщо закрити від світла фоторезистор **16** можна змінювати звук у діапазоні від 100 Гц до 3 кГц.

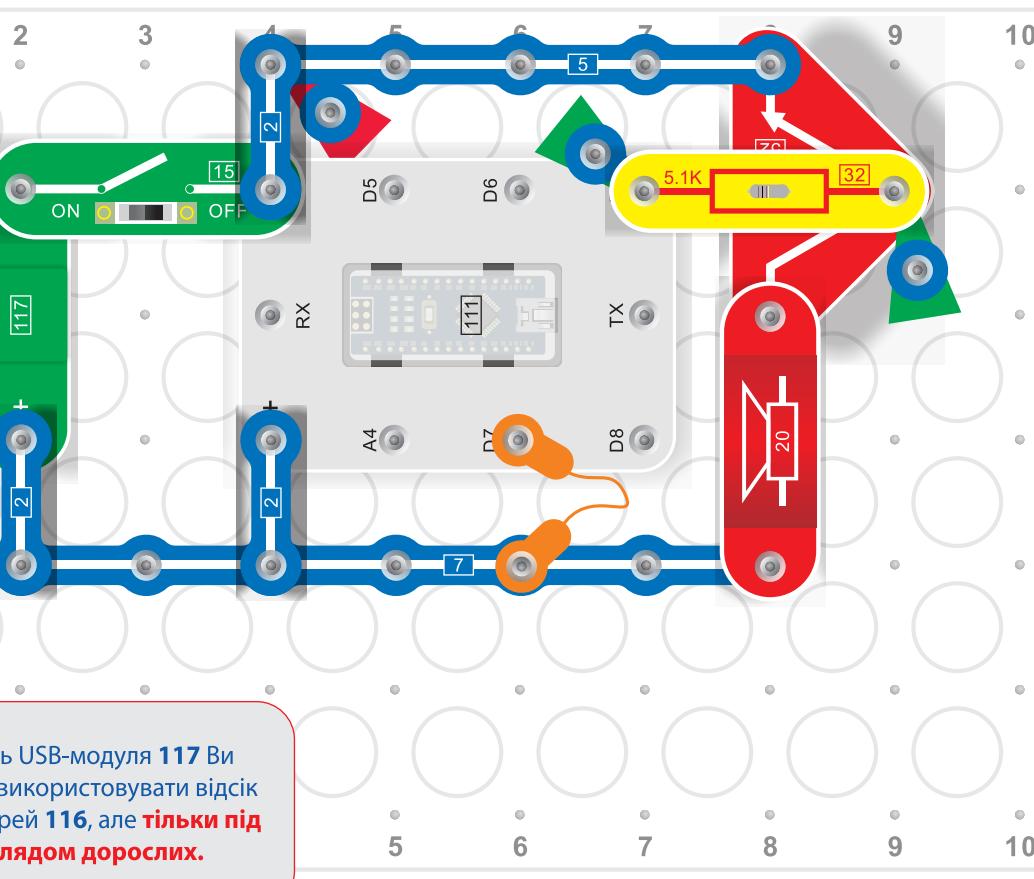
**08\_Signaling-1** — «Сигналізація, яка реагує на світло». В цій програмі сигналізація спрацює, якщо на фоторезистор **16** попаде світло. Але можна змінити програму і тоді сигналізація спрацює, якщо світло зникне. Така схема корисна для використання в теплиці, де світло повинно бути постійно.

Як можна бачити з описів програм, одна й та ж схема може виконувати абсолютно різні функції. Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження обраної програми від'єднайте кабель та увімкніть живлення – вимикач **15** в положенні ON.

## СИГНАЛІЗАЦІЇ

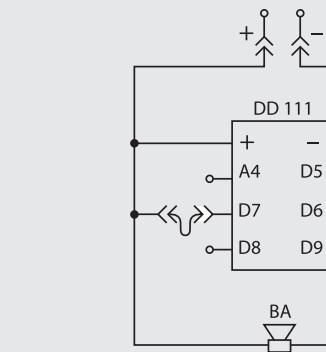


Інші типи сигналізацій описані в наступному розділі.



В автомобільних сигналізаціях при несанкціонованому відкритті двері або капоту «розривається» електричний контакт та спрацьовує сигналізація. Дуже часто в сигналізаціях замість дроту використовують геркон та магніт, які при віддалені один від одного розмикують коло. Такі схеми часто захищають вікна та двері.

Повністю програми можна подивитись на флешці, що входить до комплекту.



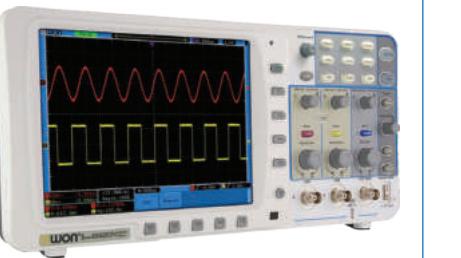
Найпростіша та найпопулярніша охоронна система, яка реагує на розрив електричного ланцюга. В даному випадку – достатньо зняти одну з клем помаранчевого дроту. Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додаються **2** програми:

**08\_Signaling-2a** — «Сирена». При розриві ланцюга пролунає звук сирени.

**08\_Signaling-2b** — «Гавкіт собаки». При розриві ланцюга пролунає гавкіт собаки.

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження програми від'єднайте кабель та увімкніть живлення – вимикач **15** у положенні ON. Від'єднайте помаранчевий дріт.

## ОСЦИЛОГРАФ



Осцилограф – прилад, який дозволяє спостерігати, вимірювати та фіксувати різноманітні коливання, наприклад, звукові. Але треба пам'ятати, що будь-яку фізичну величину, яку Ви збираєтесь досліджувати, треба попередньо перетворити на електричну напругу, оскільки осцилограф сприймає тільки її.

### 1. Меню підключення

При підключенні модулю **111 (Arduino)** до комп'ютера за допомогою USB-кабелю, послідовний СОМ-порт обирається автоматично і на кнопці з'являється напис СОМ«Х» (де Х – довільне число). Якщо цього не сталося, то варто натиснути на дану кнопку та з випадаючого списку обрати довільний СОМ-порт, після чого натиснути на кнопку «З'єднання» (п.3).

### 2. Оновити перелік портів

Якщо Ви вже спочатку запустили програму «Осцилограф», а потім під'єднали модуль **111**, то варто натиснути цю кнопку.

При натисканні цієї кнопки оновлюється перелік портів з п.1.

### 3. З'єднання

Кнопка, при натисканні якої програма «Осцилограф» намагається підключитися до модуля **111** за адресою СОМ-порту, який вказаний в Меню підключення (п.1). Якщо з'єднання не сталося, то з'явиться діалогове віно із зазначенням подальших дій.

### 4. Зберегти

При натисканні цієї кнопки в будь-який момент, поточна осцилограма буде збережена як зображення у форматі PNG. Ці химерні криві можна показувати друзям та спеціалістам!

### 5. Екран осцилографа

Частина осцилографа, на якій відображається осцилограма. По горизонтальній осі – час (sec), по вертикальній – напруга (V).

### 6. Старт

При натисканні на цю кнопку, на екрані почне відображатися осцилограма вхідного сигналу в режимі реального часу.

### 7. Стоп

При натисканні на цю кнопку, значення на екрані припиняють оновлюватися і Ви можете або зберегти їх, натиснувши кнопку «Зберегти» (п.4), або провести вимірювання за допомогою курсору миші, коли значення напруги в точці під курсором будуть відображатися на дисплеї вольтметра (п.10).

### 8. Масштаб по осі

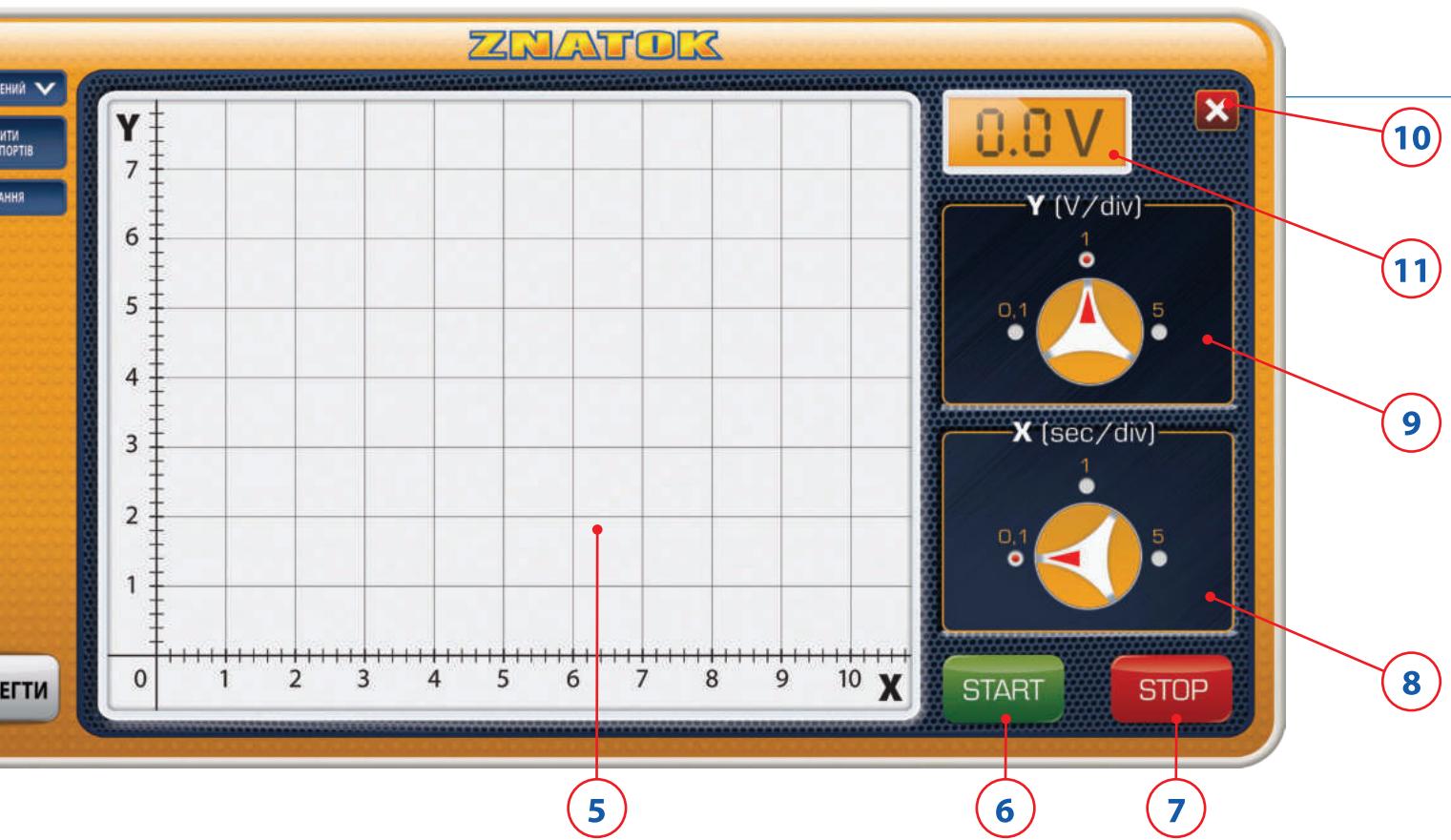
Перемикач X – змінює розгортку сигналу. Показує час, який відповідає одній оцифрованій поділці. Наприклад, якщо перемикач знаходитьться в положенні 0,1 sec/div, то це значить, що в одній поділці 0,1 секунди або 100 мілісекунд, а уся шкала дорівнює 1 секунді. Якщо в положенні 0,5 sec/div, то в одній поділці – 5 секунд, а уся шкала дорівнює 50 секундам.

### 9. Масштаб по осі Y (напруга)

Перемикач Y – змінює чутливість осцилографа. Відображає значення напруги, яке відповідає одній цифровій поділці. Наприклад, якщо перемикач знаходитьться в положенні 0,1 X/div, то це значить, що в одній поділці 0,1 вольта або 100 мВ, а уся шкала дорівнює 7 вольтам.

### 10. Вихід з програми «Осцилограф».

На даному дисплеї відображається останнє значення напруги, яке було прийняте від модуля **111**. Рекомендовано використовувати для відстеження сигналів, що повільно змінюються: схеми з реостатом або фоторезистором. Після натискання на кнопку «Стоп» (п.6), Ви можете спостерігати на дисплеї вольтметра значення напруги під курсором миші.



### 11. Вольтметр

На даному дисплеї відображається останнє значення напруги, яке було прийняте від модуля **111**. Рекомендовано використовувати для відстеження сигналів, що повільно змінюються: схеми з реостатом або фоторезистором. Після натискання на кнопку «Стоп» (п.6), Ви можете спостерігати на дисплеї вольтметра значення напруги під курсором миші.

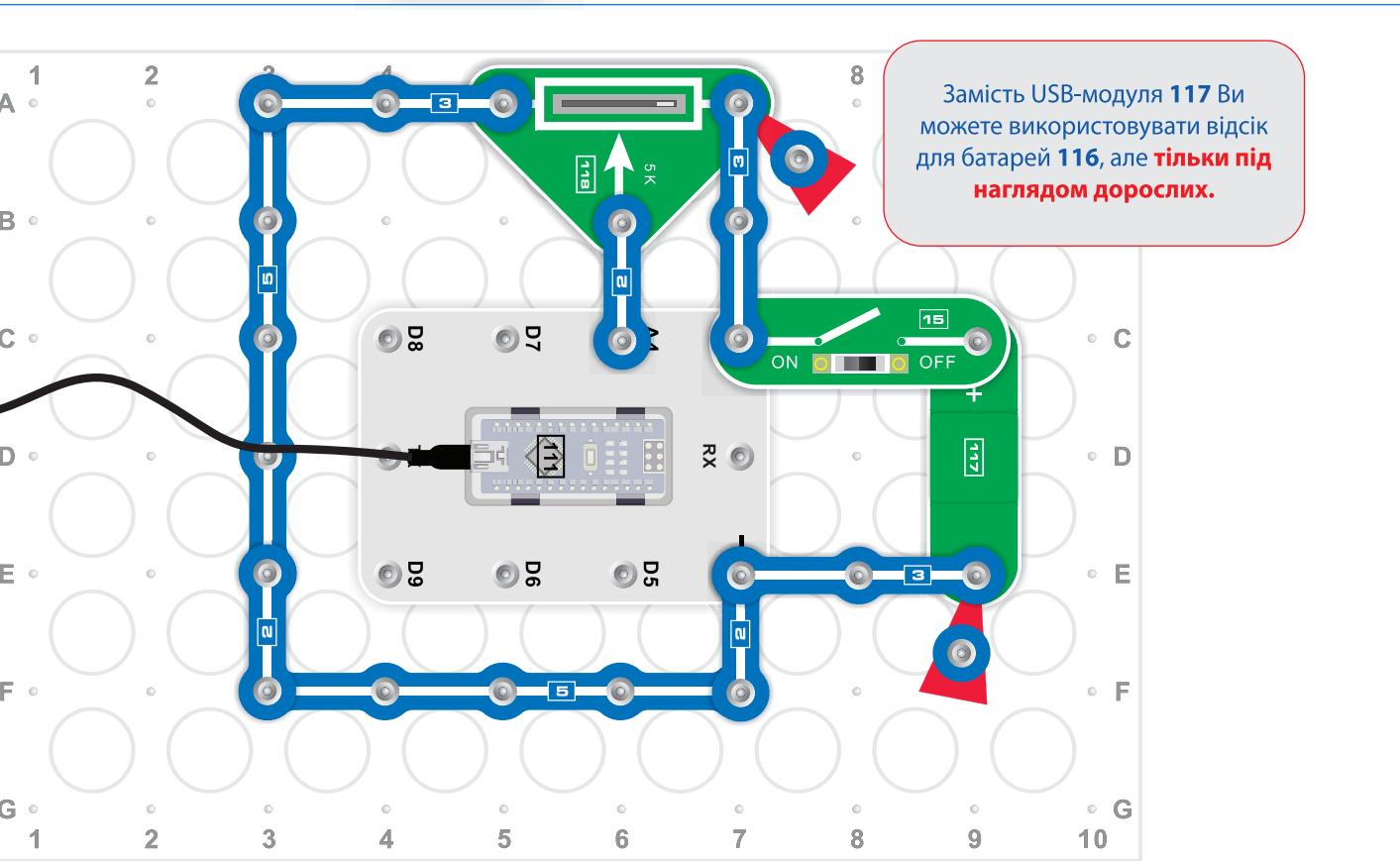
## ОСЦИЛОГРАФ (ПРОДОВЖЕННЯ)

У цій схемі Ви можете самі виступити в ролі генератора сигналів.



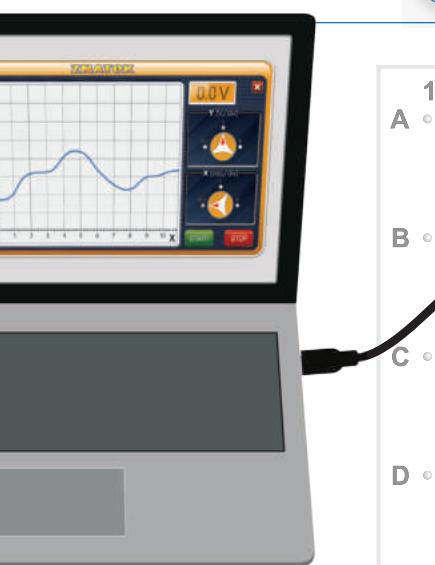
Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 1 програма: [21\\_Osc-1](#).

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження обраної програми від'єднайте кабель, під'єднайте до модуля **117** мережевий адаптер та увімкніть живлення – вимикач **15** в положенні ON. Запустіть на комп'ютері програму «Осцилограф». Регулюйте потенціал на



вході осцилографа (A4) за допомогою змінного резистора та спостерігайте за осцилограмою «сходинки», але для цього потрібно правильно підібрати розгорту осцилографа – кількість секунд у поділці (sec/div).

## ОСЦИЛОГРАФ (ЗАКІНЧЕННЯ)



В цій схемі Ви можете спостерігати та фіксувати зміну освітлення. Такі схеми корисні, наприклад, під час аналізу кількості світла, яке надходить рослинам протягом дня.

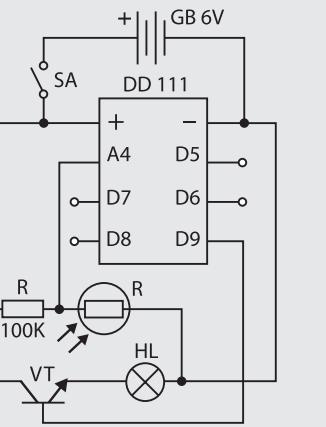
Зберіть коло, схему якого зображенено на рисунку. До цієї схеми додається 1 програма: [21\\_Osc-1](#).

Підключіть модуль **111** до комп'ютера за допомогою USB-кабелю та завантажте програму. Після завантаження обраної програми від'єднайте кабель, під'єднайте до модуля **117** мережевий адаптер та

увімкніть живлення – вимикач **15** в положенні ON. Запустіть на комп'ютері програму «Осцилограф». Закрійте фоторезистор **16** від світла та дивіться на екран.

Замість модуля **117** та мережевого блока живлення можна використовувати відсік для батарейок **116**, але дослідження можуть вийти не дуже тривалими.

## РОЗУМНЕ СВІТЛО



На жаль, досі можна бачити, як вдень на вулиці світять ліхтарі, які даремно витрачають величезну кількість електроенергії. Адже все вирішується так просто.

Зберіть коло, схему якого зображено на рисунку. До цієї схеми додаються 3 програмами:

### 23\_Smart Light-1 —

«Керування ліхтарем»

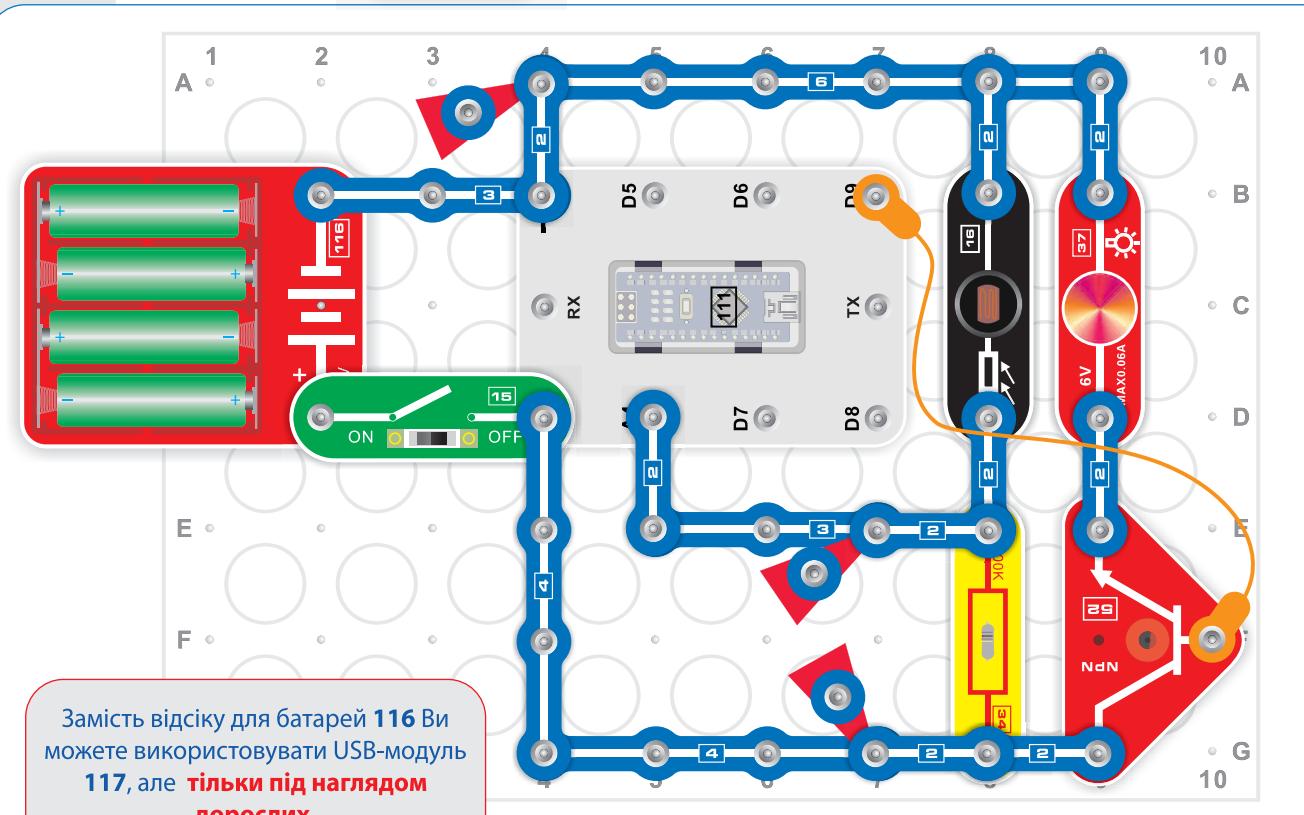
### 23\_Smart Light-5 —

«Розумна фара»

### 23\_Smart Light-6 —

«Автоматичний маяк»

Усі три програми реагують на ступінь освітленості фотодіода **16** — Лампа буде вмикатися тільки в тому випадку, коли на фотодіод потрапляє світло.



Режими роботи лампи теж усі різні: плавне увімкнення, різке увімкнення миготіння. Імітувати наступ темряви можна, якщо закривати фотодіод пальцем.

Підключіть модуль **111** до комп'ютера за

допомогою USB-кабелю та завантажте обрану програму. Після завантаження програми від'єднайте кабель та увімкніть живлення — вимикач **15** в положенні ON.

## РОЗУМНЕ СВІТЛО

### 23\_Smart Light-1

Ця програма дозволяє встановити рівень освітленості, при якому схема почне поступово збільшувати яскравість лампи. Нижче наведений фрагмент з програми та вказані параметри, за допомогою яких можна змінити порогове значення увімкнення ліхтаря, а також максимальну яскравість.

## ПРОГРАМИ

### 23\_Smart Light-5

Водії іноді забувають вмикати фари, коли настає темрява. Ця програма дозволяє робити це автоматично, а також самостійно встановлювати порогове значення освітлення, за яким фара вмикається.

### 23\_Smart Light-6

Маяк буде автоматично вмикатися з настанням сутінок. Ця програма також дозволяє встановити порогове значення освітлення, за яким лампа починає миготіти з максимальною яскравістю.

### 23\_Smart Light-6

### Порогове значення спрацювання

Змініть дане значення в межах від **0** до **1023** для регулювання чутливості сенсора

Після кожної зміни в програмі зберігайте оновлену версію з новою назвою в бібліотеці Ваших проектів. Не забувайте завантажувати оновлену програму до модуля **111**(Arduino)!

## ТОЧНІСТЬ — ЧЕМНІСТЬ КОРОЛІВ

### Чому ми так віримо цифрам та приладам?

Адже будь-який прилад НЕ є ІДЕАЛЬНИМ та у нього є багато ПОХИБОК. І усі ці похибки різні.

У цьому наборі Ви збиралі різноманітні цифрові прилади. А чи замислювались Ви, з якою точністю вони вдавали значення на індикаторі? Чому не можна одним приладом вимірювати температуру тіла людини та Сонця? Помилка в 1 мм - це багато чи мало? На ці та багато інших питань відповідає наука «Метрологія».



Можна помітити, що коли на вагах нічого немає, вони показують значення, яке відрізняється від нуля. Це значення треба віднімати від результату зважування. Така похибка називається **адитивною**.

Ваш годинник відстає на одну секунду в день. За три дні він вже буде відставати на три секунди. За 60 днів похибка помножиться, і годинник буде помилатися вже на хвилину. Така похибка називається **мультиплікативною**. Деякі прилади та перетворювачі показують абсолютно точні значення на початку та наприкінці шкали, але



Тому не дивуйтесь, якщо Ви вимірюєте температуру тіла трьома «градусниками», та отримаєте три різних значення, – вони просто мають різні похибки і, отже, характеристики.



«Наука починається там, де починаються виміри» (Д.І. Менделєєв)



## НОТИ — ЦЕ ЧАСТОТИ

У цьому наборі Ви можете відтворювати ноти (частоти), які помічені в таблиці зеленим та червоним кольорами.

Не забувайте, що майже всі сучасні музичні інструменти налаштовують за нотою **Ля** першої октави — **440 Гц**.

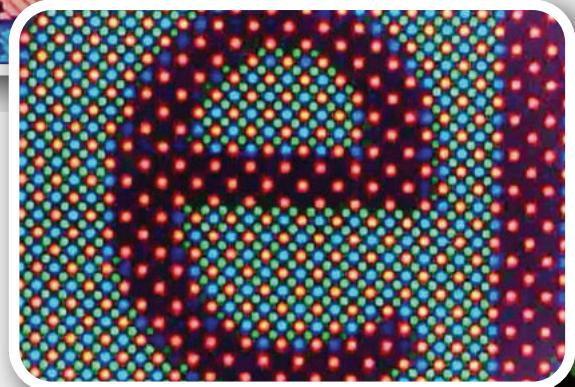
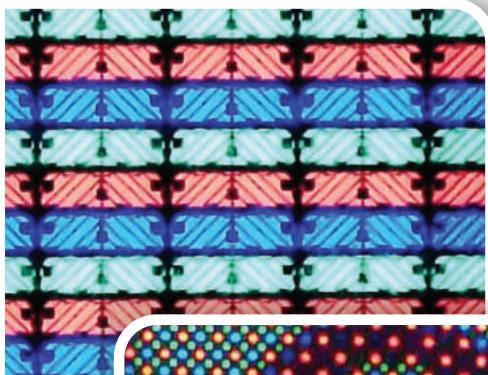
**Підвищення октави** — частота подвоюється.

**Пониження октави** — частота ділиться навпіл.

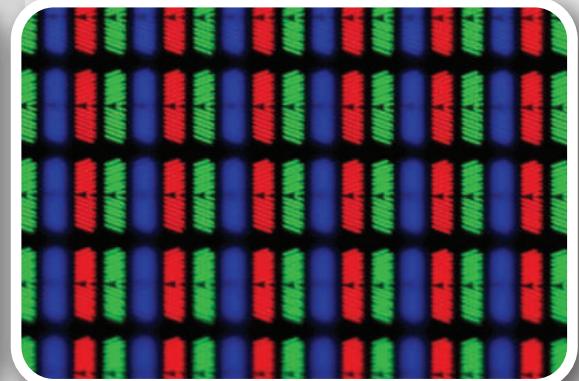
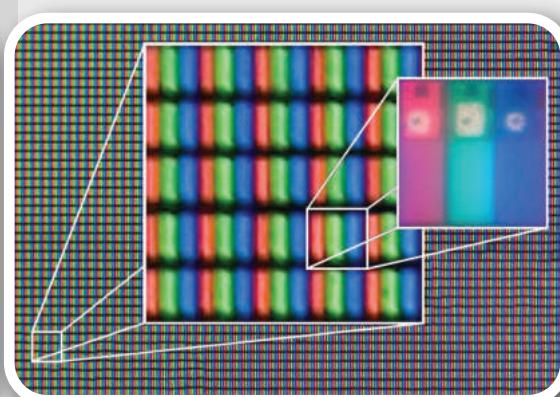
**#** — знак дієз, означає підвищення октави на півтону.



Ноти		Субконтр	Контр	Велика	Мала	Перша	Друга	Третя	Четверта	(П'ята)	(Шоста)	(Сьома)
До	С	(0)	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Ре	С#	16,35	32,70	65,41	130,82	261,63	523,25	1046,5	2093	4186	8372	16744
Ми	D	17,32	34,65	69,30	138,59	277,18	554,36	1108,7	2217,4	4434,8	8869,6	17739,2
Фа	D#	18,35	36,95	73,91	147,83	293,33	587,32	1174,6	2349,2	4698,4	9396,8	18793,6
Соль	E	19,40	38,88	77,78	155,56	311,13	622,26	1244,5	2489	4978	9956	19912
Ля	F	20,60	41,21	82,41	164,81	329,63	659,26	1318,5	2637	5274	10548	21096
	F#	21,83	43,65	87,31	174,62	349,23	698,46	1396,9	2793,8	5587,6	11175,2	22350,4
	G	23,12	46,25	92,50	185	369,99	739,98	1480	2960	5920	11840	23680
	G#	24,50	49	98	196	392	784	1568	3136	6272	12544	25088
	A	25,96	51,90	103,80	207	415,30	830,60	1661,2	3322,4	6644,8	13289,6	26579,2
	A#	27,50	55	110	220	440	880	1760	3520	7040	14080	28160
	B	29,14	58,26	116,54	233,08	466,16	932,32	1864,6	3729,2	7458,4	14916,8	29833,6

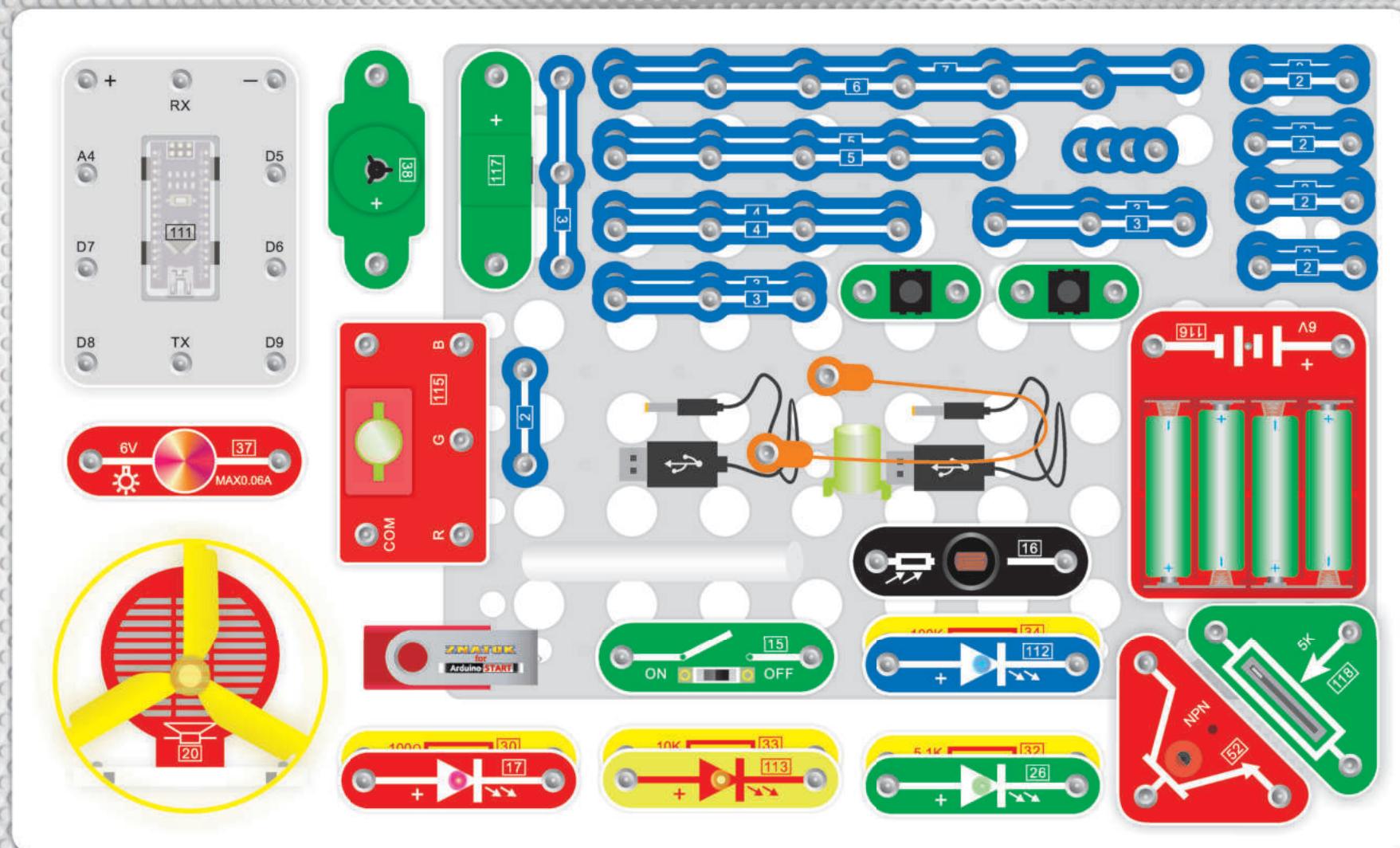
ЩО ОЗНАЧАЄ **RGB**

Уся гама кольорів, яку Ви бачите на екранах смартфонів, моніторах комп'ютерів, телевізорах, створюється за допомогою всього трьох кольорів — **червоного, зеленого та синього (RGB)**! Тут наведені фотографії різних екранів, зроблені під мікроскопом. Форма та розмір світлодіодів різні, а кольорів лише три.



# РОЗТАШУВАННЯ ЕЛЕМЕНТІВ У КОРОБЦІ

Дизайн та колір елементів може відрізнятися від наведених на малюнку



Елементи живлення не входять до комплекту